8317146

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

November 21, 2022

THIS IS TO CERTIFY THAT ANNEXED IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF THE FILE WRAPPER AND CONTENTS OF:

APPLICATION NUMBER: ***11/338,996***
FILING DATE: ***January 25, 2006***
PATENT NUMBER: ***8204134***
ISSUE DATE: ***June 19, 2012***

UNITED STATES PATENT AND TRADEMARK OFFICE · DEPARTMENT OF COMMERCE

Certified by

Kathi

Performing the Functions and Duties of the
Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

16638 U.S. PTO
012506

113260 U.S. PTO
11/338996
012506

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: HANNUKSELA

Docket: TPO.004.C1

Title: GROUPING OF IMAGE FRAMES IN VIDEO CODING

CERTIFICATE UNDER 37 CFR 1.10:
"Express Mail" mailing label number: EV 712609402 US
Date of Deposit: January 25, 2006
I hereby certify that this paper or fee is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

By: Tracey M. Dotter

CONTINUATION PATENT APPLICATION UNDER 37 C.F.R. § 1.53(b)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This is a request for filing a continuation patent application under 37 CFR § 1.53(b) of pending patent application serial number 10/306,942 filed on November 29, 2002, which claims priority under 35 U.S.C. §119 from corresponding Finnish patent application No. 20020127, filed in Finland on January 23, 2002.

We are transmitting herewith the following:

☒ Transmittal sheet containing Certificate under 37 CFR 1.10 (2 pages).
☒ Patent Application: Pages numbered 1-39, with 1pg. Abstract; 31 claims.
☒ 4 sheets of formal drawings, having Figures 1-2, 3A-3B, 4A-4C, 5A-5C, 6A-6C, and 7.
☒ Copy of the executed Declaration from the priority application (3 pages).
☒ A copy of the Change of Correspondence Address submitted in the prior pending application. Please send all future correspondence to the following address:

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

☒ The priority pending application is assigned to: Nokia Corporation at Reel/Frame No.: 013741/0536.

012506
166638 U.S. PTO

113260 U.S.PTO
11/338996
012506

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: HANNUKSELA

Docket: TPO.004.C1

Title: GROUPING OF IMAGE FRAMES IN VIDEO CODING

☒ The Power of Attorney in the prior pending application is to: Steven R. Funk, Reg. No. 37,830.

☒ Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C1) in the amount of $2,150.00 for the filing fees. Authorization is also given to charge/credit Deposit Account No. 50-3581 (TPO.004.C1) for any deficiencies/overpayments.

☒ 1 Return Postcard

The fees are calculated as follows:

Basic Filing Fee =	$ 300.00
Search Fee =	$ 500.00
Examination Fee =	$ 200.00
Fee for total # of claims over 20 = $50 * (31 – 20) =	$ 550.00
Fee for total # of independent claims over 3 = $200 * (6 – 3) =	$ 600.00
TOTAL FILING FEES DUE =	$2,150.00

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: January 25, 2006

By: _____
Steven R. Funk
Reg. No. 37,830

OIPE JC48
AUG 17 2005
PATENT & TRADEMARK OFFICE

PATENT APPLICATION

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	Miska Hannuksela	Examiner:	Unassigned
Serial No.:	10/306,942	Group Art Unit:	2125
Filed:	11/29/2002	**New Docket No.:**	**TAPA.004PA**
		Old Docket No.:	KOLS.021PA

CERTIFICATE UNDER 37 CFR 1.8: The undersigne d hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, United States Patent and Trademark O ffice, P.O. Box 1450, Alexandria, VA 22313-1450, on August 15, 2005.

By: _____

CHANGE OF CORRESPONDENCE ADDRESS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir or Madam:

Please update the attorney docket number as reflected above (**TAPA.004PA**), and change the correspondence Address for the above-referenced application as follows:

Hollingsworth & Funk, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
(952) 854-2700

If there are any questions regarding this matter, please call the undersigned at 952-854-2700.

Respectfully submitted,

By: _____

Steven R. Funk
Reg. No.: 37,830

GROUPING OF IMAGE FRAMES IN VIDEO CODING

RELATED PATENT DOCUMENTS

[0001] This application is a continuation of U.S. Patent Application Serial No. 10/306,942 filed on November 29, 2002 which is incorporated herein by reference in its entirety.

FIELD OF THE INVENTION

[0002] The invention relates to the grouping of multimedia files, particularly video files and particularly in connection with streaming.

BACKGROUND OF THE INVENTION

[0003] The term 'streaming' refers to simultaneous sending and playback of data, typically multimedia data, such as audio and video files, in which the recipient may begin data playback already before all the data to be transmitted has been received. Multimedia data streaming systems comprise a streaming server and terminal devices that the recipients use for setting up a data connection, typically via a telecommunications network, to the streaming server. From the streaming server the recipients retrieve either stored or real-time multimedia data, and the playback of the multimedia data can then begin, most advantageously almost in real-time with the transmission of the data, by means of a streaming application included in the terminal.

[0004] From the point of view of the streaming server, the streaming may be carried out either as normal streaming or as progressive downloading to the terminal. In normal streaming the transmission of the multimedia data and/or the data contents are controlled either by making sure that the bit rate of the transmission substantially corresponds to the playback rate of the terminal device, or, if the telecommunications network used in the transmission causes a bottleneck in data transfer, by making sure that the bit rate of the transmission substantially corresponds to the bandwidth available in the telecommunications network. In progressive downloading the transmission of the multimedia data and/or the data contents do not necessarily have to be interfered with at all, but the multimedia files are transmitted as such to the recipient, typically by using transfer protocol flow control. The terminals then receive, store and reproduce an exact copy of the data transmitted from the

server, which copy can then be later reproduced again on the terminal without needing to start a streaming again via the telecommunications network. The multimedia files stored in the terminal are, however, typically very large and their transfer to the terminal is time-consuming, and they require a significant amount of storage memory capacity, which is why a normal streaming is often preferred.

[0005] The video files in multimedia files comprise a great number of still image frames, which are displayed rapidly in succession (of typically 15 to 30 frames per s) to create an impression of a moving image. The image frames typically comprise a number of stationary background objects, determined by image information which remains substantially unchanged, and few moving objects, determined by image information that changes to some extent. The information comprised by consecutively displayed image frames is typically largely similar, i.e. successive image frames comprise a considerable amount of redundancy. The redundancy appearing in video files can be divided into spatial, temporal and spectral redundancy. Spatial redundancy refers to the mutual correlation of adjacent image pixels, temporal redundancy refers to the changes taking place in specific image objects in subsequent frames, and spectral redundancy to the correlation of different colour components within an image frame.

[0006] To reduce the amount of data in video files, the image data can be compressed into a smaller form by reducing the amount of redundant information in the image frames. In addition, while encoding, most of the currently used video encoders downgrade image quality in image frame sections that are less important in the video information. Further, many video coding methods allow redundancy in a bit stream coded from image data to be reduced by efficient, lossless coding of compression parameters known as VLC (Variable Length Coding).

[0007] In addition, many video coding methods make use of the above-described temporal redundancy of successive image frames. In that case a method known as motion-compensated temporal prediction is used, i.e. the contents of some (typically most) of the image frames in a video sequence are predicted from other frames in the sequence by tracking changes in specific objects or areas in successive image frames. A video sequence always comprises some compressed image frames the image information of which has not been determined using motion-compensated temporal

prediction. Such frames are called INTRA-frames, or I-frames. Correspondingly, motion-compensated video sequence image frames predicted from previous image frames, are called INTER-frames, or P-frames (Predicted). The image information of P-frames is determined using one I-frame and possibly one or more previously coded P-frames. If a frame is lost, frames dependent on it can no longer be correctly decoded.

[0008] An I-frame typically initiates a video sequence defined as a Group of Pictures (GOP), the P-frames of which can only be determined on the basis of the I-frame and the previous P-frames of the GOP in question. The next I-frame begins a new group of pictures GOP, the image information comprised by which cannot thus be determined on the basis of the frames of the previous GOP. In other words, groups of pictures are not temporally overlapping, and each group of picture can be decoded separately. In addition, many video compression methods employ bi-directionally predicted B-frames (Bi-directional), which are set between two anchor frames (I- and P-frames, or two P-frames) within a group of pictures GOP, the image information of a B-frame being predicted from both the previous anchor frame and the one succeeding the B-frame. B-frames therefore provide image information of higher quality than P-frames, but typically they are not used as anchor frames, and therefore their removal from the video sequence does not degrade the quality of subsequent images. However, nothing prevents B-frames from being used as anchor frames as well, only in that case they cannot be removed from the video sequence without deteriorating the quality of the frames dependent on them.

[0009] Each video frame may be divided into what are known as macroblocks that comprise the colour components (such as Y, U, V) of all pixels of a rectangular image area. More specifically, a macroblock consists of at least one block per colour component, the blocks each comprising colour values (such as Y, U or V) of one colour level in the image area concerned. The spatial resolution of the blocks may differ from that of the macroblocks, for example U- and V-components may be displayed using only half of the resolution of Y-component. Macroblocks can be further grouped into slices, for example, which are groups of macroblocks that are typically selected in the scanning order of the image. Temporal prediction is typically carried out in video coding methods block- or macroblock-specifically, instead of image-frame-specifically.

[0010] To allow for flexible streaming of video files, many video coding systems employ scalable coding in which some elements or element groups of a video sequence can be removed without affecting the reconstruction of other parts of the video sequence. Scalability is typically implemented by grouping the image frames into a number of hierarchical layers. The image frames coded into the image frames of the base layer substantially comprise only the ones that are compulsory for the decoding of the video information at the receiving end. The base layer of each group of pictures GOP thus comprises one I-frame and a necessary number of P-frames. One or more enhancement layers can be determined below the base layer, each one of the layers improving the quality of the video coding in comparison with an upper layer. The enhancement layers thus comprise P- or B-frames predicted on the basis of motion-compensation from one or more upper layer images. The frames are typically numbered according to an arithmetical series.

[0011] In streaming, transmission bit rate must be controllable either on the basis of the bandwidth to be used or the maximum decoding or bit rate value of the recipient. Bit rate can be controlled either at the streaming server or in some element of the telecommunications network, such as an Internet router or a base station of a mobile communications network. The simplest means for the streaming server to control the bit rate is to leave out B-frames having a high information content from the transmission. Further, the streaming server may determine the number of scalability layers to be transmitted in a video stream, and thus the number of the scalability layers can be changed always when a new group of pictures GOP begins. It is also possible to use different video sequence coding methods. Correspondingly, B-frames, as well as other P-frames of the enhancement layers, can be removed from the bit stream in a telecommunications network element.

[0012] The above arrangement involves a number of drawbacks. Many coding methods, such as the coding according to the ITU-T (International Telecommunications Union, Telecommunications Standardization Sector) standard H.263, are familiar with a procedure called reference picture selection. In reference picture selection at least a part of a P-image has been predicted from at least one other image than the one immediately preceding the P-image in the time domain. The selected reference image is signalled in a coded bit stream or in bit stream header fields image-,

image-segment- (such as a slice or a group of macroblocks), macroblock-, or block-specifically. The reference picture selection can be generalized such that the prediction can also be made from images temporally succeeding the image to be coded. Further, the reference picture selection can be generalized to cover all temporally predicted frame types, including B-frames. Since it is possible to also select at least one image preceding an I-image that begins a group of pictures GOP as the reference image, a group of pictures employing reference picture selection cannot necessarily be decoded independently. In addition, the adjusting of scalability or coding method in the streaming server or a network element becomes difficult, because the video sequence must be decoded, parsed and buffered for a long period of time to allow any dependencies between different image groups to be detected.

[0013] A further problem relates to the insertion of a video sequence in the middle of another video sequence, which has typically led to discontinuity in image numbering. The numbering of video sequence images is typically used for detecting the loss of image frames. However, if a separate video sequence, such as a commercial, is inserted into a video sequence, the separate video sequence is typically provided with separate image numbering, which is not in line with the ascending image numbering of the original video sequence. The receiving terminal may therefore interpret the deviating image numbering as a signal of lost image frames and start unnecessary actions to reconstruct the image frames suspected as lost or to request a re-transmission thereof.

[0014] A similar problem is encountered when buffering and indexing reference pictures in the encoder and the decoder. As a part of the indexing, the encoder may intentionally remove a plurality of image frames, which are unnecessary as reference image frames for the temporal prediction process of any remaining image frames. Again, there is no process for informing the decoder a reason for the removal: for example, if a plural number of image frames are removed, the receiving terminal may unnecessarily interpret these intentional removals as protocol errors.

BRIEF DESCRIPTION OF THE INVENTION

[0015] Now there is invented an improved method and equipment implementing the method, which enable the decoder to take into account the image frames the encoder has intentionally removed. Various aspects of the

invention include a method, a video encoder, a video decoder, a video signal, a device, preferably a streaming system element, for adjusting bit rate of a video sequence and a computer program that are characterized by what is stated in the independent claims.

[0016] The preferred embodiments of the invention are disclosed in the dependent claims.

[0017] The invention is based on the idea of decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding. The video sequence includes an indication relating to at least one discontinuity in the numbering of the image frames, which indication is decoded from the video sequence. Then, in response to the indication, the buffer memory is configured to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames, and the image frames in the buffer memory are used in the decoding process. Preferably, said indication informs that at least one discontinuity in the numbering of the image frames in the video sequence are intentional, and the number of image frames generated in the buffer memory are used in place of image frames that do not exist at a decoder. The missing image frames cause discontinuities in the numbering of the image frames.

[0018] According to an embodiment, after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames, the entering of the image frames comprised by the video sequence in question into the buffer memory is continued from the correct image frame number.

[0019] According to an embodiment, a number of filler frames corresponding to the discontinuities in the numbering of the image frames is entered into the buffer memory. Preferably, said filler frames are designated by an identifier indicating that the filler frames do not belong to the actual video sequence.

[0020] According to an embodiment, the step of configuring the buffer memory to provide a number of image frames corresponding to the discontinuities in the numbering of the image framesis carried out by memory indications, whereby no data is entered into the buffer memory.

[0021] An advantage of the procedure of the invention is that it provides the decoder with the information, which gaps in the image frame numbering are intentional, whereby the decoder does not start unnecessary

error correction actions. A further advantage is that the buffer memories of the encoder and the decoder can be kept in synchronism, which enables a fluent operation of the process of reference picture selection.

BRIEF DESCRIPTION OF THE DRAWINGS

[0022] In the following, the invention will be described in connection with the preferred embodiments and with reference to the accompanying drawings, in which

Figure 1 illustrates a common multimedia data streaming system in which the scalable coding hierarchy of the invention can be applied;

Figure 2 illustrates a scalable coding hierarchy of a preferred embodiment of the invention;

Figures 3a and 3b illustrate embodiments of the invention for adjusting scalability;

Figures 4a, 4b and 4c illustrate embodiments of the invention for adjusting image numbering;

Figures 5a, 5b and 5c illustrate embodiments of the invention for using B-frames in a scalable coding hierarchy;

Figures 6a, 6b and 6c illustrate scalable coding hierarchies of preferred embodiments of the invention in connection with reference picture selection; and

Figure 7 illustrates an arrangement according to a preferred embodiment of the invention for coding scene transition.

DETAILED DESCRIPTION OF THE INVENTION

[0023] In the following, a general-purpose multimedia data streaming system is disclosed, the basic principles of which can be applied in connection with any telecommunications system. Although the invention is described here with a particular reference to a streaming system, in which the multimedia data is transmitted most preferably through a telecommunications network employing a packet-switched data protocol, such as an IP network, the invention can equally well be implemented in circuit-switched networks, such as fixed telephone networks PSTN/ISDN (Public Switched Telephone Network/Integrated Services Digital Network) or in mobile communications networks PLMN (Public Land Mobile Network). Further, the invention can be applied in the streaming of multimedia files in the form of both normal

streaming and progressive downloading, and for implementing video calls, for example.

[0024] It is also to be noted that although the invention is described here with a particular reference to streaming systems and the invention can also be advantageously applied in them, the invention is not restricted to streaming systems alone, but can be applied in any video reproduction system, irrespective of how a video file that is to be decoded is downloaded and where it is downloaded from. The invention can therefore be applied for example in the playback of a video file to be downloaded from a DVD disc or from some other computer memory carrier, for example in connection with varying processing capacity available for video playback. In particular, the invention can be applied to different video codings of low bit rate that are typically used in telecommunications systems subject to bandwidth restrictions. One example is the system defined in the ITU-T standard H.263 and the one that is being defined in H.26L (possibly later to become H.264). In connection with these, the invention can be applied to mobile stations, for example, in which case the video playback can be made to adjust both to changing transfer capacity or channel quality and to the processor power currently available, when the mobile station is used also for executing other applications than video playback.

[0025] It is further to be noted that, for the sake of clarity, the invention will be described below by giving an account of image frame coding and temporal predicting on image frame level. However, in practice coding and temporal predicting typically take place on block or macroblock level, as described above.

[0026] With reference to Figure 1, a typical multimedia streaming system will be described, which is a preferred system for applying the procedure of the invention.

[0027] A multimedia data streaming system typically comprises one or more multimedia sources 100, such as a video camera and a microphone, or video image or computer graphic files stored in a memory carrier. Raw data obtained from the different multimedia sources 100 is combined into a multimedia file in an encoder 102, which can also be referred to as an editing unit. The raw data arriving from the one or more multimedia sources 100 is first captured using capturing means 104 included in the encoder 102, which capturing means can be typically implemented as different interface cards,

driver software, or application software controlling the function of a card. For example, video data may be captured using a video capture card and the associated software. The output of the capturing means 104 is typically either an uncompressed or slightly compressed data flow, for example uncompressed video frames of the YUV 4:2:0 format or motion-JPEG image format, when a video capture card is concerned.

[0028] An editor 106 links different media flows together to synchronize video and audio flows to be reproduced simultaneously as desired. The editor 106 may also edit each media flow, such as a video flow, by halving the frame rate or by reducing spatial resolution, for example. The separate, although synchronized, media flows are compressed in a compressor 108, where each media flow is separately compressed using a compressor suitable for the media flow. For example, video frames of the YUV 4:2:0 format may be compressed using the low bit rate video coding according to the ITU-T recommendation H.263 or H.26L. The separate, synchronized and compressed media flows are typically interleaved in a multiplexer 110, the output obtained from the encoder 102 being a single, uniform bit flow that comprises data of a plural number of media flows and that may be referred to as a multimedia file. It is to be noted that the forming of a multimedia file does not necessarily require the multiplexing of a plural number of media flows into a single file, but the streaming server may interleave the media flows just before transmitting them.

[0029] The multimedia files are transferred to a streaming server 112, which is thus capable of carrying out the streaming either as real-time streaming or in the form of progressive downloading. In progressive downloading the multimedia files are first stored in the memory of the server 112 from where they may be retrieved for transmission as need arises. In real-time streaming the editor 102 transmits a continuous media flow of multimedia files to the streaming server 112, and the server 112 forwards the flow directly to a client 114. As a further option, real-time streaming may also be carried out such that the multimedia files are stored in a storage that is accessible from the server 112, from where real-time streaming can be driven and a continuous media flow of multimedia files is started as need arises. In such case, the editor 102 does not necessarily control the streaming by any means. The streaming server 112 carries out traffic shaping of the multimedia data as regards the bandwidth available or the maximum decoding and playback rate

of the client 114, the streaming server being able to adjust the bit rate of the media flow for example by leaving out B-frames from the transmission or by adjusting the number of the scalability layers. Further, the streaming server 112 may modify the header fields of a multiplexed media flow to reduce their size and encapsulate the multimedia data into data packets that are suitable for transmission in the telecommunications network employed. The client 114 may typically adjust, at least to some extent, the operation of the server 112 by using a suitable control protocol. The client 114 is capable of controlling the server 112 at least in such a way that a desired multimedia file can be selected for transmission to the client, in addition to which the client is typically capable of stopping and interrupting the transmission of a multimedia file.

[0030] When the client 114 is receiving a multimedia file, the file is first supplied to a demultiplexer 116, which separates the media flows comprised by the multimedia file. The separate, compressed media flows are then supplied to a decompressor 118 where each separate media flow is decompressed by a decompressor suitable for each particular media flow. The decompressed and reconstructed media flows are supplied to a playback unit 120 where the media flows are rendered at a correct pace according to their synchronization data and supplied to presentation means 124. The actual presentation means 124 may comprise for example a computer or mobile station display, and loudspeaker means. The client 114 also typically comprises a control unit 122 that the end user can typically control through a user interface and that controls both the operation of the server, through the above-described control protocol, and the operation of the playback unit 120, on the basis of the instructions given by the end user.

[0031] It is to be noted that the transfer of multimedia files from the streaming server 112 to the client 114 takes place through a telecommunications network, the transfer path typically comprising a plural number of telecommunications network elements. It is therefore possible that there is at least some network element that can carry out traffic shaping of multimedia data with regard to the available bandwidth or the maximum decoding and playback rate of the client 114 at least partly in the same way as described above in connection with the streaming server.

[0032] Scalable coding will be described below with reference to a preferred embodiment of the invention and an example illustrated in Figure 2. Figure 2 shows part of a compressed video sequence having a first frame 200,

which is an INTRA frame, or I-frame, and thus an independently determined video frame the image information of which is determined without using motion-compensated temporal prediction. The I-frame 200 is placed on a first scalability layer, which may be referred to as an INTRA layer. Each scalability layer is assigned a unique identifier, such as the layer number. The INTRA layer may therefore be given the number 0, for example, or some other alphanumeric identifier, for example a letter, or a combination of a letter and a number.

[0033] Correspondingly, sub-sequences consisting of groups of one or more video frames are determined for each scalability layer, at least one of the images in a group (typically the first or the last one) being temporally predicted at least from a video frame of another, sub-sequence of typically either a higher or the same scalability layer, the rest of the video frames being temporally predicted either from only the video frames of the same sub-sequence, or possibly also from one or more video frames of said second sub-sequence. A sub-sequence may de decoded independently, irrespective of other sub-sequences, except said second sub-sequence. The sub-sequences of each scalability layer are assigned a unique identifier using for example consecutive numbering starting with number 0 given for the first sub-sequence of a scalability layer. Since the I-frame 200 is determined independently and can also be decoded independently upon reception, irrespective of other image frames, it also forms in a way a separate sub-sequence.

[0034] An essential aspect of the present invention is therefore to determine each sub-sequence in terms of those sub-sequences the sub-sequence is dependent on. In other words, a sub-sequence comprises information about all the sub-sequences that have been directly used for predicting the image frames of the sub-sequence in question. This information is signalled in a video sequence bit stream, preferably separate from the actual image information, and therefore the image data of the video sequence can be preferably adjusted because it is easy to determine video sequence portions that are to be independently decoded and can be removed without affecting the decoding of the rest of the image data.

[0035] Next, within each sub-sequence, the video frames of the sub-sequence are given image numbers, using for example consecutive numbering that starts with the number 0 given to the first video frame of the sub-sequence. Since the I-frame 200 also forms a separate sub-sequence, its

image number is 0. In Figure 2, the I-frame 200 shows the type (I), sub-sequence identifier and image number (0.0) of the frame.

[0036] Figure 2 further shows a next I-frame 202 of the INTRA layer, the frame thus being also an independently determined video frame that has been determined without using motion-compensated temporal prediction. The temporal transmission frequency of I-frames depends on many factors relating to video coding, image information contents and the bandwidth to be used, and, depending on the application or application environment, I-frames are transmitted in a video sequence at intervals of 0.5 to 10 seconds, for example. Since the I-frame 202 can be independently decoded, it also forms a separate sub-sequence. Since this is the second sub-sequence in the INTRA-layer, the consecutive numbering of the sub-sequence identifier of the I-frame 202 is 1. Further, since the I-frame 202 also forms a separate sub-sequence, i.e. it is the only video frame in the sub-sequence, its image number is 0. The I-frame 202 can thus be designated with identifier (I.1.0.) Correspondingly, the identifier of the next I-frame on the INTRA layer is (I.2.0.), etc. As a result, only independently determined I-frames in which the image information is not determined using motion-compensated temporal prediction are coded into the first scalability layer, i.e. the INTRA layer. The sub-sequences can also be determined using other kind of numbering or other identifiers, provided that the sub-sequences can be distinguished from one another.

[0037] The next scalability layer, which has a layer number 1, for example, and which may be referred to as the base layer, comprises coded, motion-compensated INTER or P-frames typically predicted only from previous image frames, i.e. in this case from the I-frames of an upper INTRA layer. The image information of the first P-frame 204 of the base layer shown in Figure 2 is determined using the I-frame 200 of the INTRA layer. A P-frame 204 begins the first sub-sequence of the base layer, and therefore the sub-sequence identifier of the P-frame 204 is 0. Further, since the P-frame 204 is the first image frame of the first sub-sequence of the base layer, the image number of the P-frame 204 is 0. The P-frame 204 can thus be identified with (P.0.0).

[0038] The temporally succeeding P-frame 206 of the base layer is predicted from the previous P-frame 204. The P-frames 204 and 206 thus belong to the same sub-sequence, whereby the P-frame 206 also receives the sub-sequence identifier 0. Since the P-frame 206 is the second image frame in

the sub-sequence 0, the image number of the P-frame 206 is 1, and the P-frame 206 can be identified with (P.0.1).

[0039] The scalability layer succeeding the base layer and having the layer number 2, is called enhancement layer 1. This layer comprises coded, motion-compensated P-frames predicted only from previous image frames, in this case either from I-frames of the INTRA layer or P-frames of the base layer. Figure 2 shows a first image frame 208 and a second image frame 210 of enhancement layer 1, which are both predicted only from the first image frame 200 of the INTRA layer. The P-frame 208 begins the first sub-sequence of enhancement layer 1, the sub-sequence identifier of the P-frame thus being 0. Further, since the P-frame 208 is the first and only image frame in said sub-sequence, the P-frame 208 receives the image number 0. The P-frame 208 can thus be identified with (P.0.0.).

[0040] Since the second image frame 210 is also predicted only from the first image frame 200 of the INTRA layer, the P-frame 210 begins the second sub-sequence of enhancement layer 1 and the sub-sequence identifier of the P-frame 210 is therefore 1. Since the P-frame 210 is the first image frame in the sub-sequence, the image number of the P-frame 210 is 0. The P-frame can thus be identified with (P.1.0.). The temporally succeeding P-frame 212 of enhancement layer 1 is predicted from the previous P-frame 210. The P-frames 210 and 212 thus belong to the same sub-sequence, and therefore the P-frame also receives the sub-sequence identifier 1. The P-frame 212 is the second image frame in the sub-sequence 1, and therefore the P-frame receives the image number 1 and can be identified with (P.1.1).

[0041] The temporally fourth image frame 214 of enhancement layer 1 is predicted from the first image frame 204 of the base layer. The P-frame 214 thus begins a third sub-sequence of enhancement layer 1 and therefore the P-frame 214 receives the sub-sequence identifier 2. Further, since the P-frame 214 is the first and only image frame in the sub-sequence, the image number of the P-frame 214 is 0. The P-frame 208 can therefore be identified with (P.2.0).

[0042] Also the temporally fifth image frame 216 of enhancement layer 1 is predicted only from the first image frame 204 of the base layer, the P-frame 216 thus beginning the fourth sub-sequence of enhancement layer 1, and the sub-sequence identifier of the P-frame 216 is 3. In addition, since the P-frame 216 is the first one in the sub-sequence in question, the image

number of the P-frame 216 is 0. The P-frame 216 can therefore be identified with (P.3.0). The temporally following P-frame 218 of enhancement layer 1 is predicted from the previous P-frame 216. The P-frames 216 and 218 thus belong to the same sub-sequence, and the sub-sequence identifier of the P-frame 218 is also 3. Since the P-frame 218 is the second image frame in the sub-sequence 3, the image number of the P-frame 218 is 1 and the identifier of the P-frame 218 is (P.3.1).

[0043] For simplicity and clarity of illustration the above disclosure only relates to I- and P-frames. However, a person skilled in the art will find it apparent that the scalable video coding of the invention can also be implemented using other known image frame types, such as the above described B-frames and at least SI-frames, SP-frames and MH-frames. SI-frames correspond to I-frames, but together with an SP-frame they allow an identical image to be reconstructed. An SP-frame, in turn, is a P-frame subjected to a particular coding that allows an identical image to be reconstructed together with an SI-frame or another SP-frame. SP-frames are typically placed into a video sequence into points where an access point or a scanning point is desired, or where the changing of the coding parameters of the video stream should be possible. The frames can also be used for error correction and for increasing error tolerance. SP-frames are otherwise similar to ordinary P-frames predicted from previous frames, except that they are defined so that they can be replaced by another video frame of the SP- or SI-type, the result of the decoding of the new frame being identical with the decoding result of the original SP-frame that was in the video stream. In other words, a new SP-frame that is used for replacing the one that was in the video stream is predicted from another sequence or video stream and yet the reconstructed frame has identical contents. SP-frames are described for example in the Applicant's earlier application PCT/FI02/00004.

[0044] Similarly as B-frames, macroblocks of MH (Multi Hypothesis) frames, based on motion-compensated prediction, are predicted from two other frames, which are not, however, necessarily located next to an MH-frame. More precisely, the predicted macroblocks are computed as an average of two macroblocks of two other frames. Instead of two frames, MH-frame macroblocks can naturally be also predicted from one other frame. Reference images may change according to macroblock, in other words, all macroblocks

in one and the same image are not necessarily predicted using the same frames.

[0045] A sub-sequence thus covers a specific period of time in a video sequence. The sub-sequences of the same layer or of different layers may be partly or entirely overlapping. If there are temporally overlapping image frames on the same layer, the frames are interpreted as alternative presentations of the same image content and therefore any mode of image presentation can be used. On the other hand, if there are temporally overlapping image frames on different layers, they form different presentations of the same image content, and therefore presentations differ in image quality, i.e. the quality of image is better on a lower layer.

[0046] The above disclosure referring to Figure 2 illustrates a scalable coding arrangement and a hierarchical structure and numbering of image frames according to a preferred embodiment of the invention. In this embodiment the INTRA-layer only comprises I-frames and the base layer can only be decoded using the information received from the INTRA-layer. Correspondingly, the decoding of enhancement layer 1 typically requires information from both the base layer and the INTRA-layer.

[0047] The number of the scalability layers is not restricted to three, as above, but any number of enhancement layers that is considered necessary for producing sufficient scalability may be used. Consequently, the layer number of enhancement layer 2 is four, that of enhancement layer 3 is five, etc. Since some of the image frames in the above example are given the same identifier (e.g. the identifier of both image frames 204 and 208 is (P.0.0)), by including the layer number in the identifier each image frame can be uniquely identified and, at the same time, the dependencies of each image frame on other image frames is preferably determined. Each image frame is thus uniquely identified, the identifier of image frame 204, for example, being (P.1.0.0), or simply (1.0.0) and, correspondingly, that of image 208 being (P.2.0.0), or (2.0.0).

[0048] According to a preferred embodiment of the invention, the number of a reference image frame is determined according to a specific, pre-determined alpha-numeric series, for example as an integer between 0 and 255. When the parameter value achieves the maximum value N (e.g. 255) in the series concemed, the determining of the parameter value starts from the beginning, i.e. from the minimum value of the series (e.g. 0). An image frame is

thus uniquely identified within a specific sub-sequence up to the point where the same image number is used again. The sub-sequence identifier can also be determined according to a specific, predetermined arithmetic series. When the value of the sub-sequence identifier achieves the maximum value N of the series, the determining of the identifier starts again from the beginning of the series. However, a sub-sequence cannot be assigned an identifier that is still in use (within the same layer). The series in use may also be determined in another way than arithmetically. One alternative is to assign random sub-sequence identifiers, taking into account that an assigned identifier is not be used again.

[0049] A problem in the numbering of image frames arises when the user wishes to start browsing a video file in the middle of a video sequence. Such situations occur for example when the user wishes to browse a locally stored video file backward or forward or to browse a streaming file at a particular point; when the user initiates the playback of a streaming file from a random point; or when a video file that is to be reproduced is detected to contain an error that interrupts the playback or requires the playback to be resumed from a point following the error. When the browsing of a video file is resumed from a random point after previous browsing, discontinuity typically occurs in the image numbering. The decoder typically interprets this as unintentional loss of image frames and unnecessarily tries to reconstruct the image frames suspected as lost.

[0050] According to a preferred embodiment of the invention, this can be avoided in the decoder by defining an initiation image in an independently decodable Group of Pictures GOP that is activated at the random point of the video file, and the number of the initiation image is set at zero. This independently decodable image group can thus be a sub-sequence of an INTRA-layer, for example, in which case an I-frame is used as the initiation image, or, if scaling originating from the base layer is employed, the independently decodable image group is a sub-sequence of the base layer, in which case the first image frame of the sub-sequence, typically an I-frame, is usually used as the initiation image. Consequently, when activated at a random point, the decoder preferably sets the identifier of the first image frame, preferably an I-frame, of the independently decodable sub-sequence at zero. Since the sub-sequence to be decoded may also comprise other image frames whose identifier is zero (for example when the above described alpha-

numeric series starts from the beginning), the beginning of the sub-sequence, i.e. its first image frame, can be indicated to the decoder for example by a separate flag added to the header field of a slice of the image frame. This allows the decoder to interpret the image numbers correctly and to find the correct image frame that initiates the sub-sequence from the video sequence image frames.

[0051] The above numbering system provides only one example of how the unique image frame identification of the invention can be carried out so that interdependencies between the image frames are indicated at the same time. However, video coding methods in which the method of the invention can be applied, such as video coding methods according to the ITU-T standards H.263 and H.26L, employ code tables, which in turn use variable length codes. When variable length codes are used for coding layer numbers, for example, a lower code word index, i.e. a smaller layer number, signifies a shorter code word. In practice the scalable coding of the invention will be used in most cases in such a way that the base layer will consist significantly more image frames than the INTRA-layer. This justifies the use of a lower index, i.e. a smaller layer number, on the base layer than on the INTRA-layer, because the amount of coded video data is thereby advantageously reduced. Consequently, the INTRA-layer is preferably assigned layer number 1 and the base layer is given layer number 0. Alternatively, the code can be formed by using fewer bits for coding the base layer number than the INTRA-layer number, in which case the actual layer number value is not relevant in view of the length of the code created.

[0052] Further, according to a second preferred embodiment of the invention, when the number of the scalability layers is to be kept low, the first scalability layer in particular can be coded to comprise both the INTRA-layer and the base layer. From the point of view of coding hierarchy, the simplest way to conceive this is to leave out the INTRA-layer altogether, and to provide the base layer with coded frames consisting of both independently defined I-frames, the image information of which has not been determined using motion-compensated temporal prediction, and image frames predicted from previous frames, which image frames in this case are motion-compensated P-frames predicted from the I-frames of the same layer. The layer number 0 can thus still be used for the base layer and, if enhancement layers are coded into the video

sequence, enhancement layer 1 is assigned layer number 1. This is illustrated in the following, with reference to Figures 3a and 3b.

[0053] Figure 3a shows a non-scalable video sequence structure, in which all image frames are placed on the same scalability layer, i.e. the base layer. The video sequence comprises a first image frame 300 which is an I-frame (I.0.0) and which thus initiates a first sub-sequence. The image frame 300 is used for predicting a second image frame 302 of the sub-sequence, i.e. a P-frame (P.0.1), which is then used for predicting a third image frame 304 of the sub-sequence, i.e. a P-frame (P.0.2), which is in turn used for predicting the next image frame 306, i.e. a P-frame (P.0.3). The video sequence is then provided with an I-frame (I.1.0) coded therein, i.e. an I-frame 308, which thus initiates a second sub-sequence in the video sequence. This kind of non-scalable coding can be used for example when the application employed does not allow scalable coding to be used, or there is no need for it. In a circuit-switched videophone application, for example, channel bandwidth remains constant and the video sequence is coded in real-time, and therefore there is typically no need for scalable coding.

[0054] Figure 3b, in turn, illustrates an example of how scalability can be added, when necessary, to a combined INTRA- and base layer. Here, too, the video sequence base layer comprises a first image frame 310 which is an I-frame (I.0.0) and which initiates a first sub-sequence of the base layer. The image frame 310 is used for predicting a second image frame 312 of the sub-sequence, i.e. a P-frame (P.0.1), which is then used for predicting a third image frame 314 of the sub-sequence, i.e. a P-frame (P.0.2). Enhancement layer 1, however, is also coded into this video sequence and it comprises a first sub-sequence, the first and only image frame 316 of which is a P-frame (P.0.0), which is predicted from the first image frame 310 of the base layer. The first image frame 318 of a second sub-sequence of the enhancement layer is, in turn, predicted from the second image frame 312 of the base layer, and therefore the identifier of this P-frame is (P.1.0). The next image frame 320 of the enhancement layer is again predicted from the previous image frame 318 of the same layer and, therefore, it belongs to the same sub-sequence, its identifier thus being (P.1.1).

[0055] In this embodiment of the invention the sub-sequences of the base layer can be decoded independently, although a base layer sub-sequence may be dependent on another base layer sub-sequence. The

decoding of the base layer sub-sequences requires information from the base layer and/or from the second sub-sequence of enhancement layer 1, the decoding of the sub-sequences of enhancement layer 2 requires information from enhancement layer 1 and/or from the second sub-sequence of enhancement layer 2, etc. According to an embodiment, I-frames are not restricted to the base layer alone, but lower enhancement layers may also comprise I-frames.

[0056] The basic idea behind the above embodiments is that a sub-sequence comprises information about all the sub-sequences it is dependent on, i.e. about all sub-sequences that have been used for predicting at least one of the image frames of the sub-sequence in question. However, according to an embodiment it is also possible that a sub-sequence comprises information about all sub-sequences that are dependent on the sub-sequence in question, in other words, about all the sub-sequences in which at least one image frame has been predicted using at least one image frame of the sub-sequence in question. Since in the latter case the dependencies are typically determined temporally forward, image frame buffers can be advantageously utilized in the coding in a manner to be described later.

[0057] In all the above embodiments the numbering of the image frames is sub-sequence-specific, i.e. a new sub-sequence always starts the numbering from the beginning. The identification of an individual image frame thus requires the layer number, sub-sequence identifier and image frame number to be determined. According to a preferred embodiment of the invention, the image frames can be independently numbered using consecutive numbering in which successive reference image frames in the coding order are indicated with numbers incremented by one. As regards layer numbers and sub-sequence identifiers, the above-described numbering procedure can be used. This allows each image frame to be uniquely identified, when necessary, without using the layer number and sub-sequence identifier.

[0058] This is illustrated with the example shown in Figure 4a in which the base layer comprises a temporally first I-frame 400 (I.0.0). This frame is used for predicting a first image frame 402 of enhancement layer 1, i.e. (P.0.1), which is then used for predicting a second image frame 404 belonging to the same sub-sequence (with sub-sequence identifier 0), i.e. (P.0.2), which is used for predicting a third image frame 406 of the same sub-

sequence, i.e. (P.0.3), which is used for predicting a fourth image frame 408 (P.0.4) and, finally, the fourth frame for predicting a fifth image frame 410 (P.0.5). The temporally next video sequence image frame 412 is located on the base layer, where it is in the same sub-sequence as the I-frame 400, although temporally it is only the seventh coded image frame, and therefore its identifier is (P.0.6). The seventh frame is then used for predicting a first image frame 414 of the second sub-sequence of enhancement layer 1, i.e. (P.1.7), which is then used for predicting a second image frame 416 belonging to the same sub-sequence (with sub-sequence identifier 1), i.e. (P.1.8), which in turn used for predicting a third image frame 418 (P.1.9), the third for predicting a fourth image frame 420 (P.1.10) and, finally, the fourth for predicting a fifth image frame 422 (P.1.11) of the same sub-sequence. Again, the temporally next video sequence image frame 424 is located on the base layer, where it is in the same sub-sequence as the I-frame 400 and the P-frame 412, although temporally it is only the thirteenth coded image frame and therefore its identifier is (P.0.12). For clarity of illustration, the above description of the embodiment does not comprise layer identifiers, but it is apparent that in order to implement scalability, also the layer identifier must be signalled together with the video sequence, typically as part of the image frame identifiers.

[0059] Figures 4b and 4c show alternative embodiments for grouping the image frames of the video sequence shown in Figure 4a. The image frames in Figure 4b are numbered according to sub-sequence, i.e. a new sub-sequence always starts the numbering from the beginning (from zero). Figure 4c, in turn, employs image frame numbering which corresponds otherwise to that used in Figure 4a, except that the P-frames of the base layer are replaced by SP-frame pairs to allow for identical reconstruction of image information.

[0060] As stated above, the procedure of the invention can also be implemented using B-frames. One example of this is illustrated in Figures 5a, 5b and 5c. Figure 5a shows a video sequence in the time domain, the sequence comprising P-frames P1, P4 and P7, with B-frames placed between them, the interdependencies of the B-frames with regard to temporal predicting being shown with arrows. Figure 5b shows a preferred grouping of video sequence image frames in which the interdependencies shown in Figure 5a are indicated. Figure 5b illustrates sub-sequence-specific image frame numbering in which a new sub-sequence always starts the numbering of the

image frames from zero. Figure 5c, in turn, illustrates image frame numbering, which is consecutive in the order of temporal prediction, wherein the following reference frame always receives the next image number as the previously encoded reference frame. The image frame (B1.8) (and (B2.10)) does not serve as a reference prediction frame to any other frame, therefore it does not affect the image frame numbering.

[0061] The above examples illustrate different alternatives of how scalability of video sequence coding can be adjusted by using the method of the invention. From the point of view of the terminal device reproducing the video sequence, the more scalability layers are available, or the more scalability layers it is capable of decoding, the better the image quality. In other words, increase in the amount of image information and in the bit rate used for transferring the information improves the temporal or spatial resolution, or the spatial quality of the image data. Correspondingly, a higher number of scalability layers also sets considerably higher demands on the processing capacity of the terminal device performing decoding.

[0062] In addition, the above examples illustrate the advantage gained by using sub-sequences. With image frame identifiers, the dependencies of each image frame from other image frames in the sub-sequence are indicated in an unambiguous manner. A sub-sequence thus forms an independent whole that can be left out of a video sequence, when necessary, without affecting the decoding of subsequent image frames of the video sequence. In that case, only the image frames of the sub-sequence in question and of those sub-sequences on the same and/or lower scalability layers dependent on it are not decoded.

[0063] The image frame identifier data transmitted together with the video sequence are preferably included in the video sequence header fields or in the header fields of the transfer protocol to be used for transferring the video sequence. In other words, the identifier data of predicted image frames are not included in the image data of the coded video sequence, but always into the header fields, whereby the dependencies of the image frames can be detected without decoding the images of the actual video sequence. The identifier data of the image frames can be stored for example in the buffer memory of the streaming server as the video sequence is being coded for transmission. In addition, the sub-sequences can be independently decoded on each scalability

layer, because the image frames of a sub-sequence are not dependent on other sub-sequences of the same scalability layer.

[0064] According to an embodiment of the invention, the image frames comprised by a sub-sequence may thus depend also on other sub-sequences of the same scalability layer. This dependency must then be signalled for example to the streaming server carrying out traffic shaping, because interdependent sub-sequences located on the same layer cannot be separately removed from a video sequence to be transmitted. A preferred way to carry out the signalling is to include it in the image frame identifiers to be transmitted, for example by listing the layer-sub-sequence pairs the sub-sequence in question depends on. This also provides a preferred way of indicating dependency from another sub-sequence of the same scalability layer.

[0065] The above examples illustrate a situation where image frames are temporally predicted from previous image frames. In some coding methods, however, the reference picture selection has been further extended to also include the predicting of the image information of image frames from temporally succeeding image frames. Reference picture selection offers most diversified means for creating different temporally scalable image frame structures and allows the error sensitivity of the video sequence to be reduced. One of the coding techniques based on reference picture selection is INTRA-frame postponement. The INTRA-frame is not placed into its temporally "correct" position in the video sequence, but its position is temporally postponed. The video sequence image frames that are between the "correct" position of the INTRA-frame and its actual position are predicted temporally backward from the INTRA-frame in question. This naturally requires that uncoded image frames be buffered for a sufficiently long period of time so that all image frames that are to be displayed can be coded and arranged into their order of presentation. INTRA-frame transfer and the associated determining of sub-sequences in accordance with the invention are illustrated in the following with reference to Figure 6.

[0066] Figure 6a shows a video sequence part in which the INTRA-frame comprises a single I-frame 600, which is temporally transferred to the position shown in Figure 6, although the "correct" position of the I-frame in the video sequence would have been at the first image frame. The video sequence image frames between the "correct" position and the real position 600 are thus

temporally predicted backward from the I-frame 600. This is illustrated by a sub-sequence coded into enhancement layer 1 and having a first temporally backward predicted image frame 602, which is a P-frame (P.0.0). This frame is used for temporally predicting a previous image frame 604, i.e. a P-frame (P.0.1), which is used in turn for predicting an image frame 606, i.e. a P-frame (P.0.2), and, finally, the frame 606 for predicting an image frame 608, i.e. a P-frame (P.0.3), which is at the position that would have been the "correct" position of the I-frame 600 in the video sequence. Correspondingly, the I-frame 600 on the base layer is used for temporally forward prediction of a sub-sequence comprising four P-frames 610, 612, 614 and 616, i.e. P-frames (P.0.0), (P.0.1), (P.0.2) and (P.0.3).

[0067] The fact that in this example backward predicted image frames are placed on a lower layer than forward predicted image layers indicates that for purposes of illustration, backward predicted image frames are in this coding example considered subjectively less valuable than forward predicted image frames. Naturally the sub-sequences could both be placed on the same layer, in which case they would be considered equal, or a backward predicted sub-sequence could be on the upper layer, in which case it would be considered subjectively more valuable.

[0068] Figures 6b and 6c show some alternatives for coding a video sequence according to Figure 6a. In Figure 6b both forward and backward predicted sub-sequences are placed on the base layer, the I-frame being only located on the INTRA-layer. The forward predicted sub-sequence on this layer is thus the second sub-sequence and its sub-sequence identifier is 1. In Figure 6c, in turn, an I-frame and a forward predicted sub-sequence based on it are located on the base layer, while a backward predicted sub-sequence is located on enhancement layer 1.

[0069] Moreover, according to a preferred embodiment of the invention, the above-described scalability can be utilized for coding what is known as a scene transition into a video sequence. Video material, such as news reports, music videos and movie trailers, often comprise rapid cuts between separate image material scenes. Sometimes the cuts are abrupt, but often a procedure known as scene transition is used in which transfer from one scene to another takes place by dimming, wiping, mosaic dissolving or scrolling the image frames of a previous scene, and, correspondingly, by presenting those of a later scene. From the point of view of coding efficiency,

the video coding of a scene transition is often most problematic, because the image frames appearing during the scene transition comprise information on the image frames of both the terminating and the initiating scene.

[0070] A typical scene transition, fading, is carried out by gradually reducing the intensity or luminance of the image frames of a first scene to zero, while gradually increasing the intensity of the image frames of a second scene to its maximum value. This scene transition is referred to as cross-faded scene transition.

[0071] Generally speaking, a computer-made image can be thought of as consisting of layers, or image objects. Each object can be defined with reference to at least three information types: the structure of the image object, its shape and transparency, and the layering order (depth) in relation to the background of the image and to other image objects. Shape and transparency are often determined using what is known as an alpha plane, which measures opacity and the value of which is usually determined separately for each image object, possibly excluding the background, which is usually determined as non-transparent. The alpha plane value of a non-transparent image object, such as the background, can thus be set at 1.0, whereas the alpha plane value of a fully transparent image object is 0.0. The values in between define the intensity of the visibility of a specific image object in a picture in proportion to the background and to other, at least partly overlapping, image objects that have a higher depth value than the image object in question.

[0072] The superimposition of image objects in layers according to their shape, transparency and depth position is referred to as scene composition. In practice the procedure is based on the use of weighted averages. First, the image object that is closest to the background, i.e. deepest according to its depth position, is placed onto the background and a combined image is formed of the two. The pixel values of the combined image are formed as an average weighted by the alpha plane values of the background image and the image object in question. The alpha plane value of the combined image is then set at 1.0, after which it serves as a background image for the next image object. The process continues until all image objects are attached to the image.

[0073] In the following, a procedure according to a preferred embodiment of the invention will be described in which video sequence scalability layers are combined with the above described image objects of

image frames and their information types to provide a scene transition with scalable video coding that also has good compression efficiency.

[0074] This embodiment of the invention is illustrated in the following by way of example and in a simplified manner by using cross-faded scene transition, on one hand, and abrupt scene transition, on the other hand, as examples. The image frames to be displayed during a scene transition are typically formed of two superimposed image frames, a first image frame comprising a first image scene and a second image frame a second scene. One of the image frames serves as the background image and other, which is referred to as a foreground image, is placed on top of the background image. The opacity of the background image, i.e. its non-transparency value, is constant. In other words, its pixel-specific alpha plane values are not adjusted.

[0075] In this embodiment of the invention, the background and foreground images are both defined according to scalability layer. This is illustrated in Figure 7, which shows an example of how image frames of two different scenes can be placed on scalability layers during a scene transition of the invention. Figure 7 shows a first image frame 700 of a first (terminating) scene positioned on the base layer. The image frame 700 may be either an I-frame containing image information that has not been determined using motion-compensated temporal predicting, or it may be a P-frame that is a motion-compensated image frame predicted from previous image frames. The coding of a second (initiating) scene starts during the temporally following image frame and, according to the invention, the image frames of the scene are also placed on the base layer. Remaining image frames 702, 704 of the second (terminating) scene are then placed on enhancement layer 1. These image frames are typically P-frames.

[0076] In this embodiment, the image frames of the second (initiating) scene are thus placed on the base layer, at least for the duration of the scene transition. The first image frame 706 of the scene is typically an I-frame, and it is used for temporally predicting the succeeding image frames of the scene. Consequently, the succeeding image frames of the second scene are temporally predicted frames, typically P-frames, such as frames 708 and 710 shown in Figure 7.

[0077] According to a preferred embodiment of the invention, this placing of image frames on scalability layers can be used for implementing a cross-faded scene transition by determining the image layer that is on the base

layer always as a background image of maximum opacity (100%), or non-transparency value. During a scene transition, image frames located on enhancement layers are placed onto the background image and their opacity is adjusted for example by means of suitable filters such that the frames gradually change from non-transparent to transparent.

[0078] In the video sequence of Figure 7, there are no image frames on the lower scalability layers during the first base layer image frame 700. For this time instant, the first image frame 700 is only coded into the video sequence.

[0079] The next image frame 706 of the base layer initiates a new (second) scene, during which the image frame 706 is provided with depth positioning that places it as the background image, and its opacity value is set to the maximum. Temporally simultaneously with the image frame 706 of the base layer, there is an image frame 702 of a terminating (first) scene on enhancement layer 1. To allow a cross-faded scene transition to be produced, the transparency of the frame 702 must be increased. The example of Figure 7 assumes that the opacity of the image frame 702 is set at 67% and, in addition, the image frame 702 is provided with depth positioning that determines it as a foreground image. For this time instant, an image combining the image frames 706 and 702 is coded into the video sequence, image 706 being visible as a weaker image on the background and image 702 as a stronger image at the front, because its opacity value is essentially high (67%).

[0080] During the temporally following image frame, there is a second image frame 708 of the second scene on the base layer, the frame 708 being thus correspondingly provided with depth positioning determining it as a background image, and its opacity value is set to the maximum. Enhancement layer 1 further comprises the last image frame 704 of a temporally simultaneously terminating (first) scene, the opacity value of the frame being set at 33% and, in addition, the image frame 704 being provided with depth positioning that determines it as a foreground image as well. Consequently, for this time instant, an image combined of the image frames 708 and 704 is coded into the video sequence, the image 708 being displayed as a stronger image on the background and the image 704 as a weaker image on the foreground, because the opacity value of the image 704 is no longer more than 33%.

[0081] During the temporally following image frame, the base layer comprises a third image frame 710 of the second scene. Since the first scene has terminated, only the image frame 710 is coded into the video sequence, and the displaying of the second scene continues from the frame 710.

[0082] The above disclosure describes, by way of example, the positioning of image frames according to the invention on scalability layers to implement cross-faded scene transition in a manner that is advantageous from the point of view of coding efficiency. However, it is possible that when a video sequence is being transmitted or decoded, a situation arises in which the bit rate of the video sequence must be adjusted according to the maximum value of the bandwidth and/or terminal device decoding rate available for data transfer. This kind of bit rate control causes problems when the scene transition is to be implemented using prior art video coding methods.

[0083] A preferred embodiment of the present invention now allows one or more scalability layers, or independently decodable sub-sequences included in them, to be removed from a video sequence, whereby the bit rate of the video sequence can be decreased and yet, at the same time, the video sequence can be decoded without reducing image frequency. In the image frame positioning according to Figure 7, this can be implemented by removing enhancement layer 1 from the video sequence. The video sequence is thus only used for displaying the image frames 700, 706, 708 and 710 of the base layer. In other words, a direct transition from the first (terminating) scene to the second (initiating) scene takes place in the form of an abrupt scene transition, i.e. directly from the image frame 700 of the first scene into the I-image frame 706 that initiates the second scene. The transition is thus not a cross-faded scene transition but an abrupt scene transition. Nevertheless, the scene transition can be carried out in an advantageous manner without affecting the quality of the video sequence image, and the viewer usually does not experience an abrupt scene transition carried out instead of a cross-faded scene transition in any way disturbing or faulty. On the contrary, since the prior art implementation does not allow scalability layers to be removed, scene transition would often require image frequency to be reduced, which the viewer would find jerky and disturbing.

[0084] The invention thus provides a preferred means for carrying out multimedia data traffic shaping in a streaming server comprising information about the different sub-sequences of a video sequence: their

average bit rate, location in relation to the entire video sequence, duration and their interdependencies regarding the layers. The streaming server also determines the maximum value of the bandwidth available for the data transfer and/or the decoding rate of the terminal device. On the basis of this information, the streaming server decides how many scalability layers and which sub-sequences are transmitted in the video sequence. Bit rate control can thus be carried out, when necessary, by making first a rough adjustment of the number of the scalability layers, after which finer sub-sequence-specific adjustment can be easily carried out. At its simplest, bit rate control means making sub-sequence-specific decisions on whether a particular sub-sequence will be added to a video sequence or removed from it. In case of removal it is advisable to remove entire sub-sequences from a video sequence, because the removal of separate images may cause errors in other images of the same sub-sequence. For the same reason, all sub-sequences of a lower enhancement layer should be left out if they are dependent on the removed sub-sequence of a higher layer. If there are interdependent sub-sequences on one and the same scalability layer, sub-sequences dependent on an earlier sub-sequence must be removed if the earlier sub-sequence is removed.

[0085] If the image frame identifier data are added to a video sequence that is to be transmitted, traffic shaping can also be carried out in a telecommunications network element to be used for the transfer of the video sequence, for example in an Internet router, in different gateways, or at a base station or base station controller of a mobile communications network. For the network element to be able to maintain and process the sub-sequence information, it must have extra memory and processing capacity. For this reason traffic shaping that is to be carried out in the network is perhaps most probably executed using simple processing methods, such as the DiffServ, i.e. differentiated services, procedure that is supported by some IP-based networks. In the DiffServ method, each IP data packet is assigned a priority, whereby data packets of a higher priority are relayed faster and more reliably to the recipient than packets of a lower priority. This is advantageously applied to the scalability of the invention by determining not only scalability-layer-specific, but also sub-sequence-specific priorities, which enables a highly advanced priorisation.

[0086] There are many alternatives for adding image frame identifier data to a video sequence that is to be transmitted. In addition, it is also

possible not to include any identifier data into the video sequence, in which case traffic shaping is only carried out at the streaming server. The identifier data can be included in the header fields of a video sequence, or in the header fields of the transfer protocol to be used, such as RTP (Real Time Protocol). According to a preferred embodiment, the identifier data can be transferred using a Supplemental Enhancement Information (SEI) mechanism. SEI provides a data delivery mechanism that is transferred synchronously with the video data content, thus assisting in the decoding and displaying of the video sequence. The SEI mechanism, particularly when used for transferring layer and sub-sequence information, is disclosed more in detail in the ITU-T standard document ITU-T Rec. H.264 (ISO/IEC 14496 – 10:2002), Annex D. In the cases, wherein a separate transfer protocol or mechanism is used for identifier data transfer, traffic shaping can be carried out also at one of the network elements of the transfer path. In addition, the receiving terminal device can control the decoding.

[0087] If the encoder or decoder supports reference picture selection, video sequence coding requires that decoded image frames be buffered before the coding so as to allow the relationships between different image frames to be temporally predicted from one or more other image frames. Image frame buffering can be arranged at least in two different ways, either as sliding windowing or as adaptive buffer memory control. In sliding windowing, M image frames that were coded last are used as a buffer. The frames in the buffer memory are in a decoded and reconstructed form, which allows them to be used as reference images in the coding. As the coding proceeds, the image frame buffering functions on the basis of the FIFO principle (First-In-First-Out). Images that are not used as reference images, such as conventional B-images, do not need to be stored in the buffer. Alternatively, the buffering can be also be implemented as adaptive buffer memory control, in which case the image buffering is not restricted to the FIFO principle, but image frames that are not needed can be removed from the buffer in the middle of the process, or, correspondingly, some image frames can be stored in the buffer for a longer period of time, if they are needed as reference images for later image frames. A known reference picture selection is implemented by indexing image frames that are in the buffer memory into a specific order, the image indices being then used to refer to an image in connection with motion-compensation, for example. This indexing method generally provides better compression

efficiency compared to using image numbers, for example, for referring to a specific image when motion-compensation reference images are to be signalled.

[0088] The above reference image indexing method is sensitive to transfer errors, because the buffers of the sender's encoder and the recipient's decoder must contain mutually corresponding reconstructed images in identical order to ensure that the encoder and decoder both form the same indexing order. If the image frames are indexed in different order in the buffers of the encoder and the decoder, an incorrect reference image may be used in the decoder. To prevent this, it is essential that the decoder can be controlled to take into account image frames and sub-sequences that the encoder has intentionally removed from the video sequence. In that case the image frame numbering may comprise gaps, which the decoder typically interprets as errors and tries to reconstruct the image frames interpreted as lost. For this reason, it is essential that the encoder is capable to inform the decoder that the discontinuities in the image numbering of the transmitted image frames are intentional.

[0089] In response to this, and provided that sliding windowing is used for buffering the image frames, the decoder enters into the buffer memory a number of image frames, the contents of which may be fully random, corresponding to the missing image numbers. These random image frames are then designated by an identifier "invalid" to indicate that the frames in question do not belong to the actual video sequence, but are only filler frames entered for purposes of buffer memory management. A filler frame can naturally be implemented using only memory indicators, i.e. no data is preferably entered into the buffer memory, but memory management is used merely to store a reference to a generic "invalid" frame. The entering of the image frames of the actual video sequence continues from the correct image frame number after the number of filler frames indicated by the missing image numbers has been entered into the buffer, which allows the buffer memories of the encoder and the decoder to be kept preferably in synchronism. If during decoding a reference to an image number is detected which is then found to indicate a filler frame located in the buffer, error correction actions are initiated in the decoder to reconstruct the actual reference image, for example by asking the encoder to re-transmit the reference image in question.

[0090] Further, the procedure of the invention allows separate buffer memories to be used on the different scalability layers, or, correspondingly, sub-sequence-specifically. Each scalability layer may thus have a separate buffer memory that is conceptually separate and functions on the basis of the sliding window principle. Similarly, each sub-sequence may also be provided with a conceptually separate buffer memory that also functions on the basis of the sliding window principle. This means that the buffer memory is always emptied when a sub-sequence terminates. Separate buffer memories can be used in a preferred manner for reducing the need for signalling in certain situations in which ordinary sliding window buffering would be inadequate and actively adaptive buffer memory management would need to be used instead.

[0091] The H.26L standard defines a picture order count as a picture position in output order. The decoding process specified in the H.26L standard uses picture order counts to determine default index orderings for reference pictures in B slices, to represent picture order differences between frames and fields for vector scaling in motion vector prediction and for implicit mode weighted prediction in B slices, and to determine when successive slices in decoding order belong to different pictures. The picture order count is coded and transmitted for each picture.

[0092] In one embodiment of the invention, the decoder uses the picture order count to conclude that pictures are temporally overlapping, i.e., pictures that have an equal picture order count are temporally overlapping. Preferably, the decoder outputs only the picture on the highest received layer. In the absence of layer information, the decoder concludes that the latest temporally overlapping picture in decoding order resides on highest received layer.

[0093] The above disclosure describes a procedure for coding video frames for the purpose of producing a scalable, compressed video sequence. The actual procedure is carried out in a video encoder, such as the compressor 108 of Figure 1, which may be any known video encoder. For example a video encoder according to the ITU-T recommendation H.263 or H.26L may be used, the video encoder being arranged to form, in accordance with the invention, a first sub-sequence into a video sequence, at least part of the sub-sequence being formed by coding I-frames; to form at least a second sub-sequence into the video sequence, at least part of the sub-sequence being formed by coding at least P- or B-frames, and at least one video frame of

the second sub-sequence being predicted from at least one video frame of the first sub-sequence; and to determine into the video sequence the identification data of at least the video frames of the second sub-sequence.

[0094] According to the procedure of the invention, each sub-sequence of a particular scalability layer is preferably independently decodable, naturally taking into account dependencies from higher scalability layers and possibly other sub-sequences of the same scalability layer. A scalably compressed video sequence such as the one described above can thus be decoded by decoding a first sub-sequence of a video sequence, at least part of the sub-sequence having been formed by coding at least I-frames, and by decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence having been formed by coding at least P- or B-frames, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence, and by determining the identification and dependency data of at least the video frames comprised by the second sub-sequence of the video sequence, and by reconstructing at least part of the video sequence on the basis of the sub-sequence dependencies.

[0095] The actual decoding takes places in the video decoder, such as the decompressor 118 of Figure 1, which may be any known video decoder. For example, a low bit rate video decoder according to the ITU-T recommendation H.263 or H.26L may be used, which in this invention is arranged to decode a first sub-sequence of a video sequence, at least part of the sub-sequence having been formed by coding I-frames; to decode at least a second sub-sequence of the video sequence, at least part of the second sub-sequence having been formed by coding at least P- or B-frames and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence. The video decoder is arranged to determine the identification and dependency data of at least the video frames comprised by the second sub-sequence of the video sequence and to reconstruct at least part of the video sequence on the basis of the dependencies of the sub-sequences.

[0096] An essential aspect in the operation of the streaming system of the invention is that the encoder and decoder are positioned at least so that the encoder is operationally connected to the streaming server and the decoder is operationally connected to the receiving terminal device. However,

the different elements of the streaming system, terminal devices in particular, may include functionalities that allow two-way transfer of multimedia files, i.e. transmission and reception. The encoder and decoder can thus be implemented in the form of what it known as a video codec integrating both encoder and decoder functionalities.

[0097] It is to be noted that according to the invention the functional elements of the above described streaming system and its elements, such as the streaming server, video encoder, video decoder and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two. The coding and decoding methods of the invention are particularly suitable for implementation as computer software comprising computer-readable commands for executing the process steps of the invention. A preferred way of implementing the encoder and the decoder is to store them in a storage means as a program code that can be executed by a computer-like device, for example a personal computer (PC) or a mobile station, to provide coding/decoding functionalities on the device in question.

[0098] Another alternative is to implement the invention as a video signal comprising a scalably compressed video sequence which in turn comprises video frames coded according to at least a first and a second frame format, the video frames according to the first frame format being independent of other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames. According to the invention, the video signal in question comprises at least a first sub-sequence, at least part of which has been formed by coding at least video frames of the first frame format; at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format; and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and at least one data field that determines video frames belonging to the second sub-sequence.

[0099] It is apparent to a person skilled in the art that as technology advances the basic idea of the invention can be implemented in various ways. The invention and its embodiments are therefore not restricted to the above examples, but they may vary within the scope of the claims.

CLAIMS

1. A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising:

decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames;

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames; and

using the image frames in the buffer memory in the decoding process.

2. A method according to claim 1, wherein said indication informs that at least one discontinuity in the numbering of the image frames in the video sequence are intentional.

3. A method according to claim 1, further comprising

using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

4. A method according to claim 1, wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.

5. A method according to claim 1, further comprising

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames.

6. A method according to claim 1, further comprising

entering into the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames.

7. A method according to claim 6, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the actual video sequence.

8. A method according to claim 1, wherein

the step of configuring the buffer memory to provide a number of image frames corresponding to the discontinuities in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.

9. A method according to claim 3, further comprising

detecting an error in the compressed video sequence on the basis of a reference in the decoding process to an image frame corresponding to a discontinuity in the numbering of image frames.

10. A video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding, the video decoder being arranged to

decode from the video sequence an indication related to at least one discontinuity of numbering of the image frames;

configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames; and

use the image frames in the buffer memory in the decoding process.

11. A video decoder according to claim 10, wherein

said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional.

12. A video decoder according to claim 10, wherein the video decoder is further arranged to

use the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

13. A video decoder according to claim 10, wherein

the number of image frames in the buffer memory are used in sliding window buffering of image frames.

14. A video decoder according to claim 10, wherein the video decoder is further arranged to

continue the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of

image frames corresponding to the discontinuities in the numbering of the image frames.

15. A video decoder according to claim 10, wherein the video decoder is further arranged to

enter into the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames.

16. A video decoder according to claim 15, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the actual video sequence.

17. A video decoder according to claim 10, wherein the video decoder is further arranged to

configure the buffer memory to provide a number of image frames corresponding to the discontinuities in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

18. A computer program product, stored on a computer readable medium and executable in a data processing device, for decoding a compressed video sequence, image frames of said video sequence being entered into a buffer memory in connection with the decoding, the computer program product comprising:

a computer program code for decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames;

a computer program code for configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames; and

a computer program code for including the image frames in the buffer memory in the decoding process.

19. A computer program product according to claim 18, wherein

said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional.

20. A computer program product according to claim 18, the computer program product further comprising:

a computer program code for including the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

21. A computer program product according to claim 18, the computer program product further comprising

a computer program code for using the number of image frames in the buffer memory in sliding window buffering of image frames.

22. A computer program product according to claim 18, the computer program product further comprising

a computer program code for continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames.

23. A computer program product according to claim 18, the computer program product further comprising:

a computer program code for entering into the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames.

24. A computer program product according to claim 23, the computer program product further comprising:

a computer program code for designating said filler frames by an identifier indicating that the filler frames do not belong to the actual video sequence.

25. A computer program product according to claim 18, the computer program product further comprising:

a computer program code for configuring the buffer memory to provide a number of image frames corresponding to the discontinuities in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

26. A video encoder for encoding a video sequence, the video encoder being arranged to

encode the video sequence such that at least one image frame is excluded from the video sequence, causing a discontinuity in the numbering of image frames, and

encode into the video sequence an indication related to the discontinuity of numbering of the image frames.

27. A video encoder according to claim 26, wherein

said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional.

28. A device for adjusting bit rate of a video sequence, the device being arranged to

receive an encoded video sequence;

exclude at least one image frame from the video sequence, thereby causing a discontinuity in the numbering of image frames; and

include into the video sequence an indication related to the discontinuity of numbering of the image frames.

29. A device according to claim 28, wherein

said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional.

30. A video signal comprising a compressed video sequence comprising image frames; and

an indication related to the discontinuity of numbering of the image frames in the transmitted video signal,

wherein the indication informs a decoder to use in a decoding process a number of image frames corresponding to the discontinuity in the numbering of the image frames.

31. A video signal according to claim 30, wherein

said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional.

ABSTRACT

A method of decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding. The video sequence includes an indication relating to at least one discontinuity in the numbering of the image frames, which indication is decoded from the video sequence. Then, in response to the indication, the buffer memory is configured to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames, and the image frames provided by the buffer memory are used in the decoding process. Preferably, said indication informs that at least one discontinuity in the numbering of the image frames in the video sequence are intentional, and the number of image frames provided by the buffer memory are used in place of image frames that do not exist at a decoder.

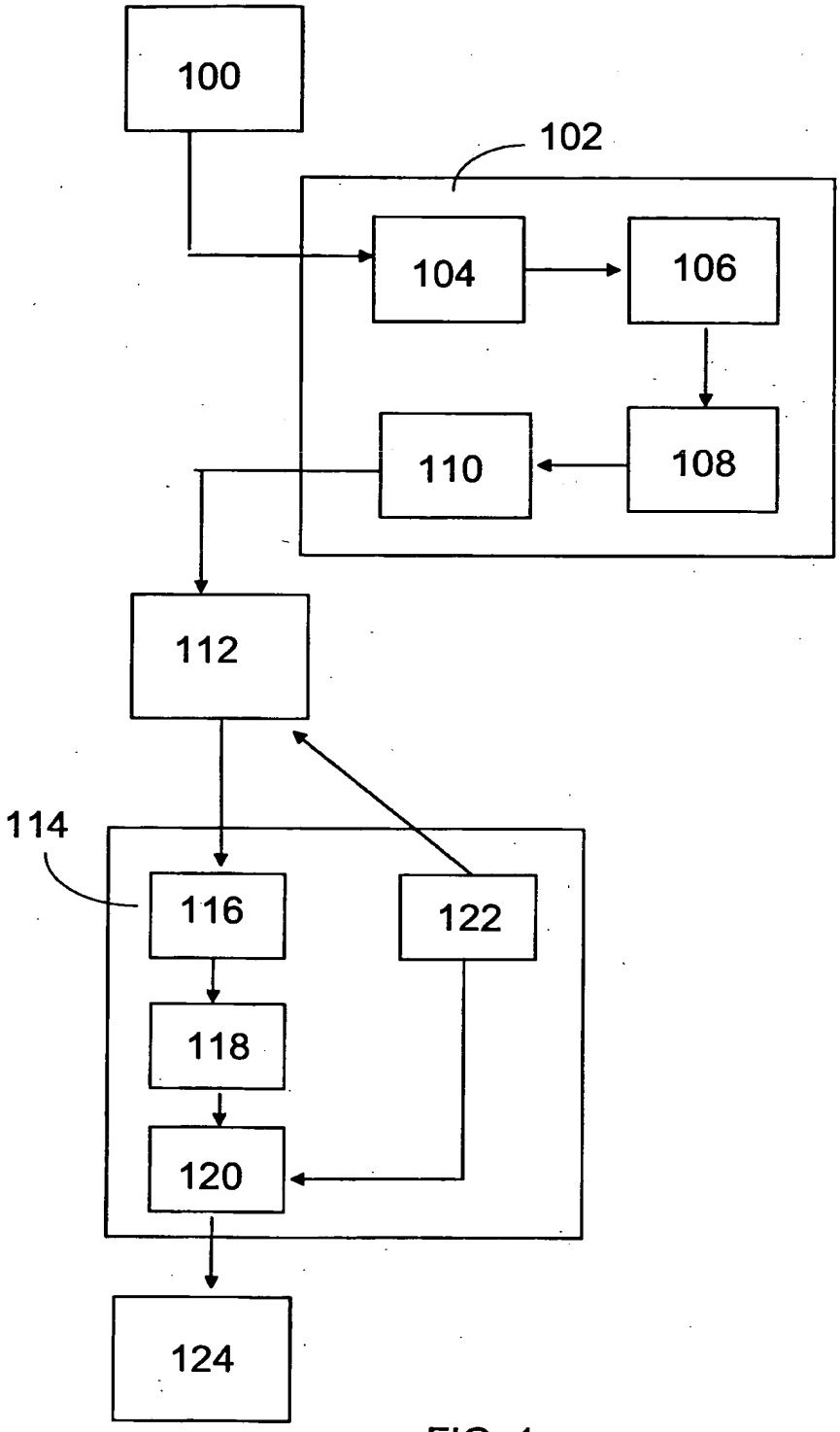


FIG. 1

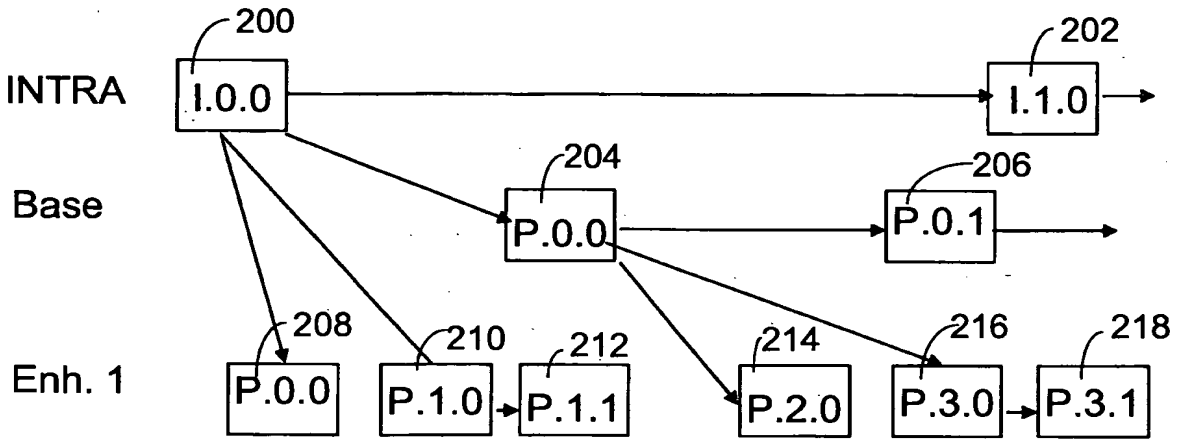


FIG. 2

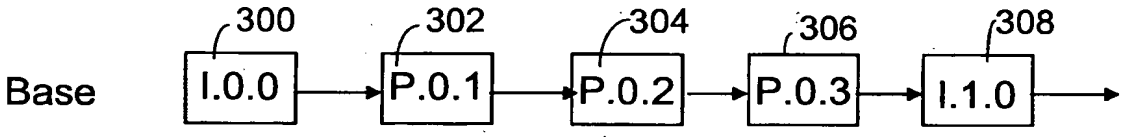


FIG. 3a

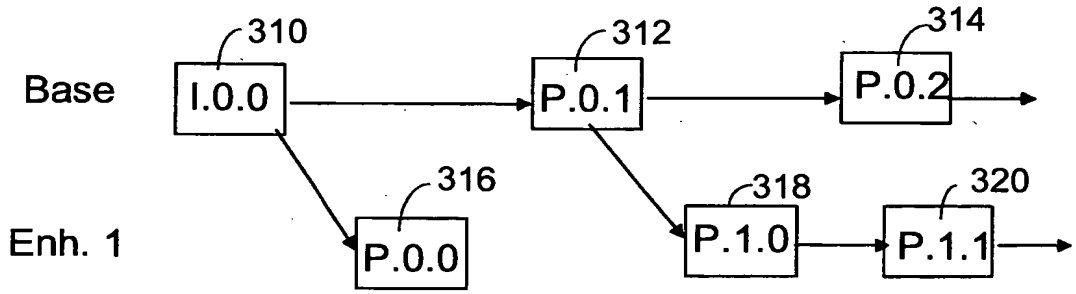


FIG. 3b

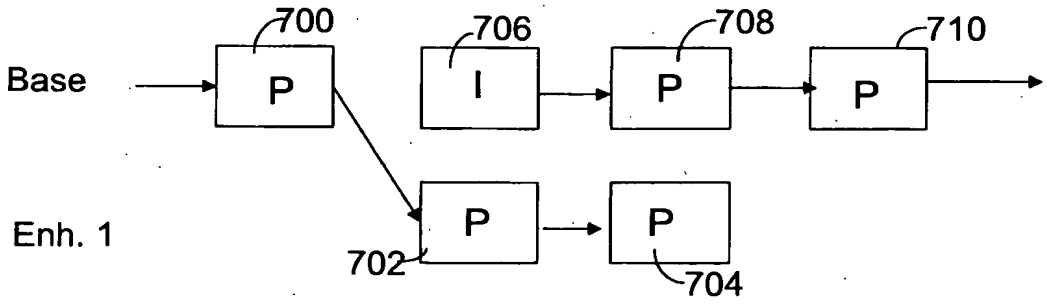


FIG. 7

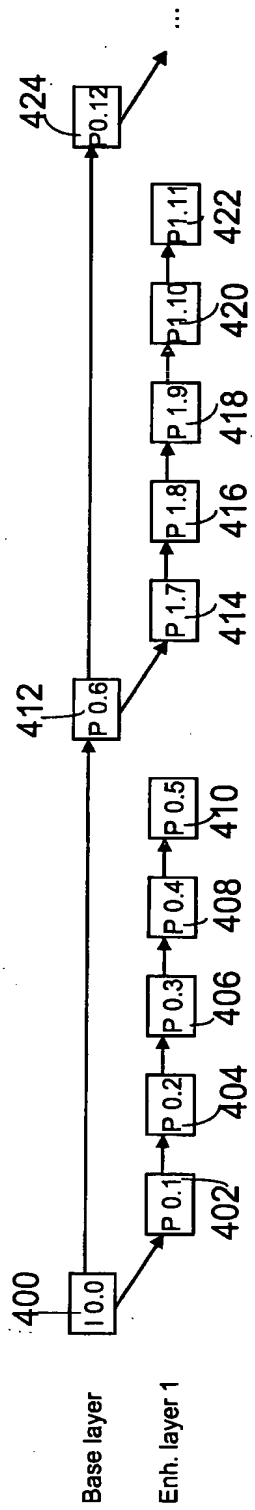


FIG. 4a

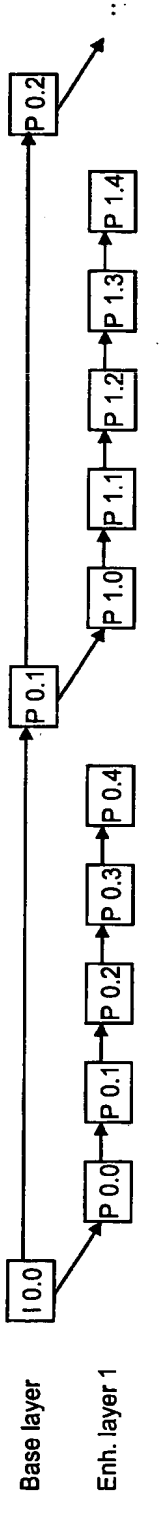


FIG. 4b

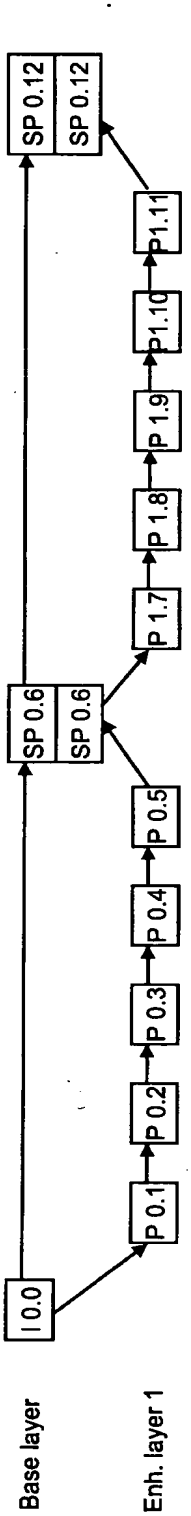


FIG. 4c

FIG. 5a

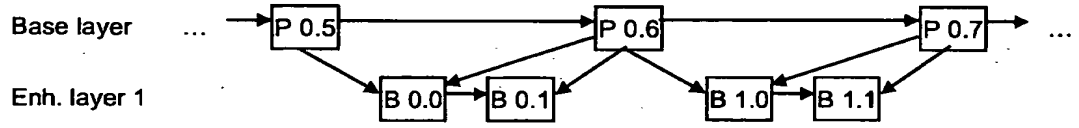


FIG. 5b

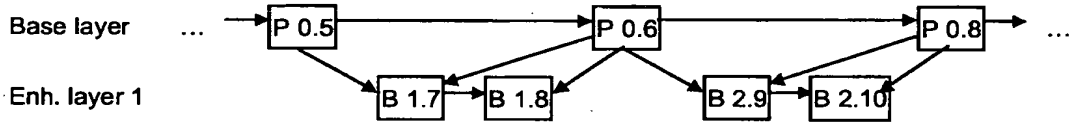


FIG. 5c

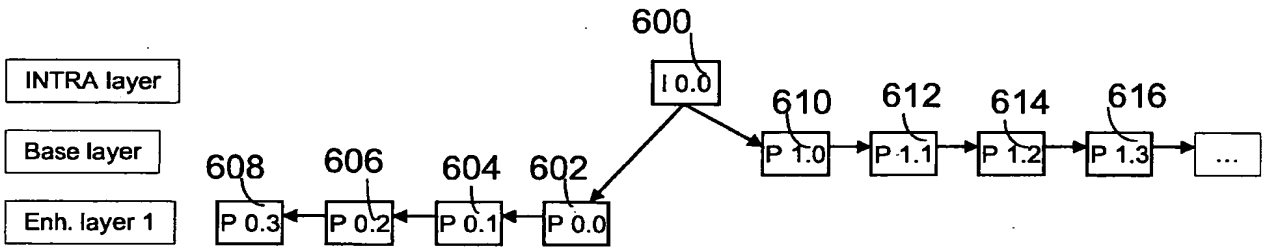


FIG. 6a

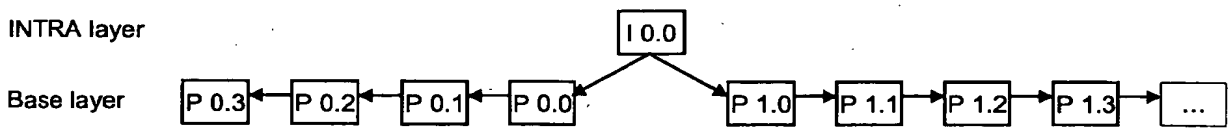


FIG. 6b

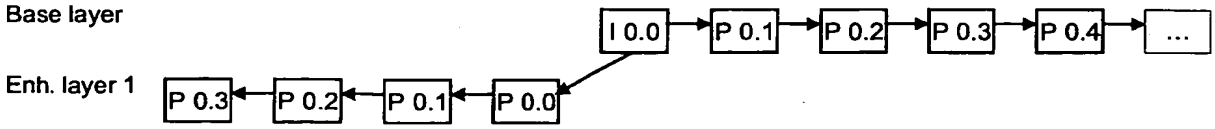


FIG. 6c

CRAWFORD MAUNU PLLC

United States Patent Application

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below named inventor I hereby declare that: my residence, post office address and citizenship are as stated below next to my name; that

I verily believe I am the original; first and sole inventor (if only one name is listed below) or a joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: **GROUPING OF IMAGE FRAMES IN VIDEO CODING.**

The specification of which

a. ☐ is attached hereto

b. ☒ is entitled **GROUPING OF IMAGE FRAMES IN VIDEO CODING,** having attorney docket number {KOLS.021PA (NC 35273US)}.

c. ☒ was filed on November 29, 2002 as application serial no. 10/306,942 and was amended on (if applicable) (in the case of a PCT-filed application) described and claimed in international no. filed and as amended on (if any), which I have reviewed and for which I solicit a United States patent.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, § 1.56 (attached hereto).

I hereby claim foreign priority benefits under Title 35, United States Code, § 119/365 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on the basis of which priority is claimed:

a. ☐ no such applications have been filed.

b. ☒ such applications have been filed as follows:

FOREIGN APPLICATION(S), IF ANY, CLAIMING PRIORITY UNDER 35 USC § 119			
COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	DATE OF ISSUE (day, month, year)
FINLAND	20020127	23 JANUARY 2002	
ALL FOREIGN APPLICATION(S), IF ANY, FILED BEFORE THE PRIORITY APPLICATION(S)			
COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	DATE OF ISSUE (day, month, year)

I hereby claim the benefit under Title 35, United States Code, § 120/365 of any United States and PCT international application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. APPLICATION NUMBER	DATE OF FILING (day, month, year)	STATUS (patented, pending, abandoned)

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below:

U.S. PROVISIONAL APPLICATION NUMBER	DATE OF FILING (Day, Month, Year)

I hereby appoint the following attorney(s) and/or patent agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith:

Funk, Steven R.	Reg. No. 37,830	Crawford, Robert J.	Reg. No. 32,122	Maunu, LeRoy D.	Reg. No. 35,274
Hollingsworth, Mark A.	Reg. No. 38,491	Curtin, Eric J.	Reg. No. 47,511	Davis, Clara	Reg. No. 50,495

I hereby authorize them to act and rely on instructions from and communicate directly with the person/assignee/attorney/firm/organization who/which first sends/sent this case to them and by whom/which I hereby declare that I have consented after full disclosure to be represented unless/until I instruct Crawford PLLC.

Please direct all correspondence in this case to Crawford PLLC at the address indicated below:

Crawford PLLC
1270 Northland Drive, Suite 390
St. Paul, Minnesota 55120

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

2	Full Name Of Inventor	Family Name HANNUKSELA	First Given Name MISKA	Second Given Name
0	Residence & Citizenship	City TAMPERE	State or Foreign Country FINLAND	Country of Citizenship FINLAND
1	Post Office Address	Post Office Address KUKKANIITYNKATU 4 B	City TAMPERE	State & Zip Code/Country 33710/FINLAND
Signature of Inventor 201:			Date:	7 January 2003

§ 1.56 Duty to disclose information material to patentability.

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is canceled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is canceled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§ 1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) the closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made of record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

- A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application:

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.

BEST AVAILABLE COPY

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

01/27/2006 MAHMED1 00000032 503581 11338996

01 FC:1011	300.00	DA
02 FC:1111	500.00	DA
03 FC:1311	200.00	DA
04 FC:1201	600.00	DA
05 FC:1202	550.00	DA

PTO-1556
(5/87)

*U.S. Government Printing Office: 2002 — 490-267/69033

PTO/SB/06 (12-04)
Approved for use through 7/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875 Effective December 8, 2004

Application or Docket Number: 11/338,996

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	150.00		N/A	300.00
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A	N/A	$250		N/A	$500
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	$100		N/A	$200
TOTAL CLAIMS (37 CFR 1.16(i))	31 minus 20 =	* 11	X$ 25 =		OR	X$50 =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	6 minus 3 =	* 3	X100 =			X200 =	
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))			+180=			+360=	
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X$ 25 =		OR	X$50 =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X100 =		OR	X200 =	
Application Size Fee (37 CFR 1.16(s))									
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					+180=		OR	+360=	
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X$ 25 =		OR	X$50 =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X100 =		OR	X200 =	
Application Size Fee (37 CFR 1.16(s))									
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					+180=		OR	+360=	
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPE IAP2 JUN 19 2006 PATENT & TRADEMARK OFFICE

IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,996	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on June 13, 2006.

By Rennae Johnson
Rennae Johnson

INFORMATION DISCLOSURE STATEMENT (37 C.F.R. §1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted either within three months of the filing date or before the first Office Action of the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

In accordance with 37 C.F.R. §1.98(a)(2), and the 5 August 2003 Official Gazette Notice, only a copy of each foreign document or other non-patent published document listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

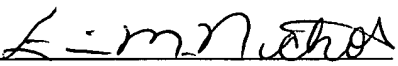
No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. §1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP §609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: June 13, 2006

By: 
Erin M. Nichols
Reg. No. 57,125

FORM 1449

INFORMATION DISCLOSURE STATEMENT

IN AN APPLICATION

(Use several sheets if necessary)

OIPE IAP2 JUN 19 2006 PATENT & TRADEMARK OFFICE

Docket Number: TPO.004.C1	Application Number: 11/338,996
Applicant: HANNUKSELA	
Filing Date: 01/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5,699,476	16.12.1997	VAN DER MEER			
	5,774,593	30.6.1998	ZICK ET AL.			
	6,108,382	22.8.2000	GRINGERI ET AL.			
	6,307,886	23.10.2001	WESTERMANN			
	6,337,881	8.1.2002	CHADDHA			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	WO 00/67469	9.11.2000	PCT			n/a	
	WO 01/47283	28.06.2001	PCT			n/a	
	WO 02/054776	11.7.2002	PCT			n/a	
	EP 1 005 232	31.05.2000	EP			n/a	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	9-17 Oct. 2002	Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, *Editor's Proposed Draft Text Modification for Joint Video Specification (ITU-T Rec. 264 ISO/IEC 14496-10 AVC)*, Geneva modifications draft 37.
	15 Jan. 1998	ITU-Telecommunications Standardization Union, *Draft Text of Recommendation H.263 Version 2 ("H.263+") for Decision.*
	7/10/02	ITU-Telecommunications Standardization Union, *H.26L Test Model Long Term Number 8 (TML-8) Draft0*, Document VCEG-N10. pp. 1-46.
	Nov. 2000	ITU-Telecommunications Standardization Union, *Draft for "H.263++" Annexes U,V, and W to Recommendation H.263.*
	12 Nov. 2002	David Singer and Toby Walker, *Study Text of ISO/IEC 14496-15/FCD,* pp. 1-25.
	12 Nov. 2002	David Singer and Toby Walker, *Study Text of ISO/IEC 14496-15/FCD (Revised),* 26 pages.
	Sept. 2002	Hannuksela, M et al.; Sub-Picture: ROI Coding and Unequal Error Protection, IEEE 2002 Conference, September 2002, USA (internet publication).
	Sept. 2002	Hannuksela, M et al.; Sub-Picture: Video-Coding for Unequal Error Protection, XI European Signal Processing Conference, September 2002, Toulouse, France (internet publication).
	7.12.2001	Wang, Y.K. et al.; Core Experiment Description of Sub-Picture Coding, ITU 15. Meeting 3.- 7.12.2001, Pattaya, Thailand (internet publication)
	2001	B.A. Lokshin, "Digital Broadcasting: from Studio to Audience" Cyrus Systems, Moscow, 2001. (translation included)

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification: **H04N 5/00, H04N 7/26**	**A1**	(11) International Publication Number:	**WO 00/67469**
		(43) International Publication Date:	09 November 2000 (09.11.2000)

(21) International Application Number: PCT/FI00/00363

(22) International Filing Date: 26 April 2000 (26.04.2000)

(30) Priority Data:
990970 29 April 1999 (29.04.1999) FI

(60) Parent Application or Grant
NOKIA MOBILE PHONES LTD. [/]; (). HANNUKSELA, Miska [/]; (). WALKER, Andrew ; ().

Published

(54) Title: DATA TRANSMISSION
(54) Titre: TRANSMISSION DE DONNEES

(57) Abstract

A data transmission system (10) is provided for transmitting a data signal between a server (12, 14, 16, 18) and a communications terminal (20, 22) over a transmission link having a variable bandwidth. The data signal is a scalable compressed multimedia clip comprising a sequence of images. Each image has a base layer and a number of enhancement layers. To cope with variability in the bandwidth, the sequence is re-ordered so that the base layers have greater safety times than the enhancement layers. This allows all of the base layers to arrive in time to be played at the expense of the enhancement layers.

(57) Abrégé

L'invention concerne un système de transmission de données (10) qui assure la transmission d'un signal de données entre un serveur (12, 14, 16, 18) et un terminal de communication (20, 22) sur une liaison à largeur de bande variable. Le signal de données est un signal écrêté multimédia comprimé variable comprenant une séquence d'images. Chaque image comporte une couche de base et un certain nombre de couches d'accentuation. Pour tenir compte de la variabilité de la largeur de bande, on réordonne la séquence pour que la couche de base dispose de temps de sécurité accrus par rapport aux couches d'accentuation. Cela permet aux couches de base de se présenter à temps pour être sollicitées aux dépens des couches d'accentuation.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04N 5/00, 7/26 — A1

(11) International Publication Number: WO 00/67469

(43) International Publication Date: 9 November 2000 (09.11.00)

(21) International Application Number: PCT/FI00/00363

(22) International Filing Date: 26 April 2000 (26.04.00)

(30) Priority Data:
990970 29 April 1999 (29.04.99) FI

(71) Applicant: NOKIA MOBILE PHONES LTD. [FI/FI]; Keilalahdentie 4, FIN–02150 Espoo (FI).

(72) Inventor: HANNUKSELA, Miska; Kukkaniitynkatu 4 B, FIN–33710 Tampere (FI).

(74) Agent: WALKER, Andrew; Nokia Corporation, P.O. Box 319, FIN–00045 Nokia Group (FI).

(81) Designated States: AE, AG, AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), DM, DZ, EE, EE (Utility model), ES, FI, FI (Utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

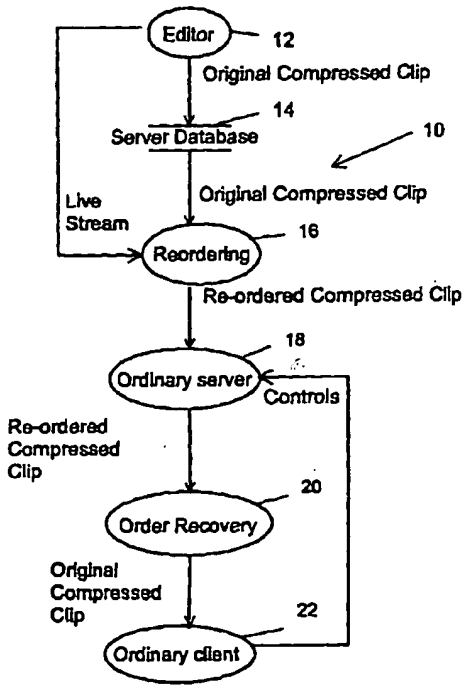
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: DATA TRANSMISSION

(57) Abstract

A data transmission system (10) is provided for transmitting a data signal between a server (12, 14, 16, 18) and a communications terminal (20, 22) over a transmission link having a variable bandwidth. The data signal is a scalable compressed multimedia clip comprising a sequence of images. Each image has a base layer and a number of enhancement layers. To cope with variability in the bandwidth, the sequence is re–ordered so that the base layers have greater safety times than the enhancement layers. This allows all of the base layers to arrive in time to be played at the expense of the enhancement layers.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Description

DATA TRANSMISSION

This invention relates to data transmission and is particularly, but not exclusively, related to transmission of data representative of picture sequences, such as video. It is particularly suited to transmission over links susceptible to errors and loss of data.

During the past few years, the amount of multimedia content available through the Internet has increased considerably. Since data delivery rates to mobile terminals are becoming high enough to enable such terminals to be able to retrieve multimedia content, it is becoming desirable to enable mobile terminals to retrieve video and other multimedia content from the Internet. An example of a high-speed data delivery system is the upcoming GSM phase 2+.

The term multimedia as used herein applies to both sound and pictures, to sound only and to pictures only. Sound may include speech and music.

Network traffic through the Internet is based on a transport protocol called the Internet Protocol (IP). IP is concerned with transporting data packets from one location to another. It facilitates the routing of packets through intermediate gateways, that is, it allows data to be sent to machines that are not directly connected in the same physical network. The unit of data transported by the IP layer is called an IP datagram. The delivery service offered by IP is connectionless, that is IP datagrams are routed around the Internet independently of each other. Since no resources are permanently committed within the gateways to any particular connection, the gateways may occasionally have to discard datagrams because of lack of buffer space or other resources. Thus, the delivery service offered by IP is a best effort service rather than a guaranteed service.

Internet multimedia is typically streamed over the User Datagram Protocol (UDP), the Transmission Control Protocol (TCP) or the Hypertext Transfer Protocol (HTTP).

UDP is a connectionless lightweight transport protocol. It offers very little above the service offered by IP. Its most important function is to deliver datagrams between specific transport endpoints. Consequently, the transmitting application has to take care of how to packetize data to datagrams. Headers used in UDP contain a checksum that allows the UDP layer at the receiving end to check the validity of the data. Otherwise, degradation of IP datagrams will in turn affect UDP datagrams. UDP does not check that the datagrams have been received, does not retransmit missing datagrams, nor does it guarantee that the datagrams are received in the same order as they were transmitted.

UDP introduces a relatively stable throughput having a small delay since there are no retransmissions. Therefore it is used in retrieval applications to deal with the effect of network congestion and to reduce delay (and jitter) at the receiving end. However, the client must be able to recover from packet losses and possibly conceal lost content. Even with reconstruction and concealment, the quality of a reconstructed clip suffers somewhat. On the other hand, playback of the clip is likely to happen in real-time without annoying pauses. Firewalls, whether in a company or elsewhere, may forbid the usage of UDP because it is connectionless.

TCP is a connection-orientated transport protocol and the application using it can transmit or receive a series of bytes with no apparent boundaries as in UDP. The TCP layer divides the byte stream into packets, sends the packets over an IP network and ensures that the packets are error-free and received in their correct order. The basic idea of how TCP works is as follows. Each time TCP sends a packet of data, it starts a timer. When the receiving end gets the packet, it immediately sends an acknowledgement back to the sender. When the sender receives the acknowledgement, it knows all is well, and cancels the timer. However, if the IP layer loses the outgoing segment or the return acknowledgement, the timer at the sending end will expire. At this point, the sender will retransmit the segment. Now, if the sender waited for an acknowledgement for each packet before sending the next one, the overall

transmission time would be relatively long and dependent on the round-trip delay between the sender and the receiver. To overcome this problem, TCP uses a sliding window protocol that allows several unacknowledged packets to be present in the network. In this protocol, an acknowledgement packet contains a field filled with the number of bytes the client is willing to accept (beyond the ones that are currently acknowledged). This window size field indicates the amount of buffer space available at the client for storage of incoming data. The sender may transmit data within the limit indicated by the latest received window size field. The sliding window protocol means that TCP effectively has a slow start mechanism. At the beginning of a connection, the very first packet has to be acknowledged before the sender can send the next one. Typically, the client then increases the window size exponentially. However, if there is congestion in the network, the window size is decreased (in order to avoid congestion and to avoid receive buffer overflow). The details how the window size is changed depend on the particular TCP implementation in use.

A multimedia content creation and retrieval system is shown in Figure 1. The system has one or more media sources, for example a camera and a microphone. Alternatively, multimedia content can also be synthetically created without a natural media source, for example animated computer graphics and digitally generated music. In order to compose a multimedia clip consisting of different media types, such as video, audio, text, images, graphics and animation, raw data captured from the sources are edited by an editor. Typically the storage space taken up by raw (uncompressed) multimedia data is huge. It can be megabytes for a video sequence which can include a mixture of different media, for example animation. In order to provide an attractive multimedia retrieval service over low bit rate channels, for example 28.8 kbps and 56 kbps, multimedia clips are compressed in the editing phase. This typically occurs off-line. The clips are then handed to a multimedia server. Typically, a number of clients can access the server over one or more networks. The server is able to respond to the requests presented by the clients. The main task of the server is to transmit a desired multimedia clip to the client which the client decompresses and plays. During

playback, the client utilizes one or more output devices, such as a screen and a loudspeaker. In some circumstances, clients are able to start playback while data are still being downloaded.

It is convenient to deliver a clip by using a single channel which provides a similar quality of service for the entire clip. Alternatively different channels can be used to deliver different parts of a clip, for example sound on one channel and pictures on another. Different channels may provide different qualities of service. In this context, quality of service includes bit rate, loss or bit error rate and transmission delay variation.

In order to ensure multimedia content of a sufficient quality is delivered, it is provided over a reliable network connection, such as TCP, which ensures that received data are error-free and in the correct order. Lost or corrupted protocol data units are retransmitted. Consequently, the channel throughput can vary significantly. This can even cause pauses in the playback of a multimedia stream whilst lost or corrupted data are retransmitted. Pauses in multimedia playback are annoying.

Sometimes retransmission of lost data is not handled by the transport protocol but rather by some higher-level protocol. Such a protocol can select the most vital lost parts of a multimedia stream and request the retransmission of those. The most vital parts can be used for prediction of other parts of the stream, for example.

In order to understand the invention better, descriptions of the elements of the retrieval system, namely the editor, the server and the client, are set out below.

A typical sequence of operations carried out by the multimedia clip editor is shown in Figure 2. Raw data are captured from one or more data sources. Capturing is done using hardware, device drivers dedicated to the hardware and a capturing application which controls the device drivers to use the hardware. Capturing hardware may consist of a video camera connected to a PC video grabber card,

for example. The output of the capturing phase is usually either uncompressed data or slightly compressed data with irrelevant quality degradations when compared to uncompressed data. For example, the output of a video grabber card could be in an uncompressed YUV 4:2:0 format or in a motion-JPEG format. The YUV colour model and the possible sub-sampling schemes are defined in Recommendation ITU-R BT.601-5 "Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios". Relevant digital picture formats such as CIF, QCIF and SQCIF are defined in Recommendation ITU-T H.261 "Video Codec for Audiovisual Services at p x 64 kbits" (section 3.1 "Source Formats").

During editing separate media tracks are tied together in a single timeline. It is also possible to edit the media tracks in various ways, for example to reduce the video frame rate. Each media track may be compressed. For example, the uncompressed YUV 4:2:0 video track could be compressed using ITU-T recommendation H.263 for low bit rate video coding. If the compressed media tracks are multiplexed, they are interleaved so that they form a single bitstream. This clip is then handed to the multimedia server. Multiplexing is not essential to provide a bitstream. For example, different media components such as sounds and images may be identified with packet header information in the transport layer. Different UDP port numbers can be used for different media components.

A typical sequence of operations carried out by the multimedia server is shown in Figure 3. Typically multimedia servers have two modes of operation; they deliver either pre-stored multimedia clips or a live (real-time) multimedia stream. In the first mode, clips are stored in a server database, which is then accessed on-demand by the server. In the second mode, multimedia clips are handed to the server as a continuous media stream that is immediately transmitted to clients. Clients control the operation of the server by an appropriate control protocol being at least able to select a desired media clip. In addition, servers may support more advanced controls. For example, clients may be able to stop the transmission of a clip, to pause and resume transmission of a clip, and to control the media flow in

case of a varying throughput of the transmission channel in which case the server must dynamically adjust the bitstream to fit into the available bandwidth.

A typical sequence of operations carried out by the multimedia retrieval client is shown in Figure 4. The client gets a compressed and multiplexed media clip from a multimedia server. The client demultiplexes the clip in order to obtain separate media tracks. These media tracks are then decompressed to provide reconstructed media tracks which are played out with output devices. In addition to these operations, a controller unit is provided to interface with end-users, that is to control playback according to end-user input and to handle client-server control traffic. It should be noted that the demultiplexing-decompression-playback chain can be done on a first part of the clip while still downloading a subsequent part of the clip. This is commonly referred to as streaming. An alternative to streaming is to download the whole clip to the client and then demultiplex it, decompress it and play it.

A typical approach to the problem of varying throughput of a channel is to buffer media data in the client before starting the playback and/or to adjust the transmitted bit rate in real-time according to channel throughput statistics.

One way of tackling the problem of pauses is by using dynamic bit rate adjustment in the multimedia server. However, the server may not react to network congestion sufficiently quickly to avoid pausing in the client. In addition, the server cannot control the retransmission mechanism of TCP (or other underlying protocols, such as IP).

Even if dynamic bit rate adjustment is used, the client has to do some initial buffering in any case to avoid delivery delays caused by re-transmission. If a constant channel bit rate is assumed, one can calculate the point in time at which a data unit is supposed to have been completely received. In addition, one can calculate the point in time by which a data unit is supposed to have been played. The time difference between these two points in time is referred to as the safety

time. Another way of defining the safety time is to state that it is the maximum time between two consecutively received data units which does not cause pausing in the playback.

When calculating the safety times for a clip, each data unit has to be considered separately. The calculations assume that no throughput drops occur before the data unit that is currently being processed. If the maximum throughput of the channel is equal to the average bit rate of the multimedia clip, the client cannot recover from a drop in the amount of received bits after throughput has dropped. The only way to guarantee some protection against throughput drops is to buffer some data before starting playback. If the channel stops providing data, the client can still play the stream while there are data in a buffer. Thus, the average safety time is approximately equal to the initial buffering time. Since the bit rate of the clip may vary, the safety time also may vary and the minimum safety time is thus equal to or less than the initial buffering delay.

HTTP, the Hypertext Transfer Protocol, is the basis of the World Wide Web (WWW). It is a simple protocol. A client establishes a TCP connection to a server, issues a request, and reads back the server's response. The server denotes the end of its response by closing the connection. The arrangement of protocol layers is typically HTTP on TCP which is on IP.

The most common HTTP request is called GET. The GET request is associated with a universal resource identifier (URI) which uniquely specifies the requested item. The server responds to the GET request by returning the file corresponding to the specified URI. The file returned by the server normally contains pointers (hypertext links) to other files that can reside on other servers. A user can therefore easily follow the links from file to file.

Servers used for Internet multimedia retrieval are either dedicated multimedia servers or normal WWW servers.

Dedicated multimedia servers are typically capable of transmission over HTTP, TCP and UDP protocols. They may be able to readjust the contents of media clips dynamically to meet the available network bandwidth and to avoid network congestion. They may also support fast forward and fast rewind operations as well as live multimedia streaming. They can provide a number of streams simultaneously.

Multimedia servers based on normal WWW servers are also referred to as server-less or HTTP multimedia solutions. The multimedia clips are streamed over HTTP. Since this type of server has no control over the contents of the stream, no flow (bandwidth) control can be applied, and it cannot respond to network congestion. Therefore, sudden pauses in the playback can occur. Consequently there must be a relatively long initial buffering delay in the client before starting the playback to avoid such sudden pauses. Fast-forwarding a multimedia stream from a standard WWW server is not possible. Live multimedia streaming must be implemented using special tricks, such as Java programming.

When a streamed multimedia clip is received, a suitable independent media player application or a browser plug-in can be used to play it. Such multimedia players differ largely from browser to browser. Newer browsers may have some integrated plug-ins for the most popular streaming video players.

There are a number of different data transmission methods available for transmitting data between mobile terminals and their networks. One of the best known methods is GSM (Global System for Mobile communications).

The current GSM data service called Circuit Switched Data (CSD) offers a 9.6 kbps circuit-switched channel. It is intended for GSM to provide a 14.4 kbps data channel having forward error correction (FEC) and status information. High Speed Circuit Switched Data (HSCSD) provides multiple 9.6 kbps or 14.4 kbps time slots for a single user at the same time. There are symmetric and asymmetric connections. In a symmetric connection, the air interface resources are allocated

symmetrically, and provide the same data transmission rate in both directions. In an asymmetric connection, different data rates are supported for up-link and for down-link. However, asymmetric air interface connections are applicable only in non-transparent mode (see below).

Circuit-switched GSM data systems, CSD and HSCSD, offer two basic connection types, namely transparent (T) and non-transparent (NT). These are distinguished by the way they correct transmission errors. In a transparent connection, error correction is done solely by a forward error correction mechanism provided by the radio interface transmission scheme. The connection is considered as a synchronous circuit. The available throughput is constant, and the transmission delay is fixed. The transmitted data are likely to contain bit inversion errors. In a non-transparent connection, the GSM circuit connection is regarded as a packet (or frame) data flow although the end-to-end service is circuit-switched. Each frame includes redundancy bits to enable a receiver to detect remaining errors. There are two sources of error, packet drops and corruption due to interference in the radio frequency path. The latter can be recovered by redundancy checking. The Radio Link Protocol (RLP) is used to provide retransmission in case of remaining errors in frame. If a frame is found to be correct, the receiver acknowledges this fact. If it is found not to be correct, a negative acknowledgement is sent and the indicated frame is retransmitted. Consequently, a non-transparent connection is error-free but throughput and transmission delay vary.

Other network types exist, for example GPRS (General Packet Radio System). In GPRS the transmissions are truly packet based.

A video sequence consists of a series of still pictures. Video compression methods are based on reducing redundant and perceptually irrelevant parts of video sequences. The redundancy in video sequences can be categorised into spatial, temporal and spectral redundancy. Spatial redundancy means the correlation between neighbouring pixels. Temporal redundancy means that the same objects

appear in consecutive pictures. Reducing the temporal redundancy reduces the amount of data required to represent a particular image sequence and thus compresses the data. This can be achieved by generating motion compensation data, which describe the motion between the current and a previous (reference) picture. In effect, the current picture is predicted from the previous one. Spectral redundancy means the correlation between the different colour components of the same picture.

Simply reducing the redundancy of the sequence does not usually compress it enough. Therefore, some video encoders try to reduce the quality of those parts of a video sequence which are subjectively the least important. In addition, the redundancy of the encoded bitstream is reduced by means of efficient lossless coding of compression parameters and coefficients. The main technique is to use variable length codes.

Video compression methods typically differentiate between pictures that can use temporal redundancy reduction and those that cannot. Compressed pictures, which do not use temporal redundancy reduction methods, are usually called INTRA or I-frames whereas temporally predicted pictures are called INTER or P-frames. In the INTER frame case, the predicted (motion-compensated) picture is rarely precise enough, and therefore a spatially compressed prediction error picture is also associated with each INTER frame.

Temporal scalability provides a mechanism for enhancing perceptual quality by increasing the picture display rate. This is achieved by taking a pair of consecutive reference pictures and bi-directionally predicting a B-picture from either one or both of them. The B-picture can then be played in sequence between the two anchor pictures. This is illustrated in Figure 5. Bi-directional temporal prediction yields a more accurately predicted picture than uni-directional prediction. Thus, for the same quantization level, B-pictures yield increased compression as compared to forwardly predicted P-pictures. B-pictures are not used as reference pictures, that is other pictures are never predicted from them. Since they can be discarded

without impacting the picture quality of future pictures, they provide temporal scalability. It should be noted that while B-pictures provide better compression performance than P-pictures, they are more complex to construct and require more memory. Furthermore they introduce additional delays because bi-directional interpolation requires both reference pictures to have been received and additional calculations are required. In addition, B-pictures require more side information in the bitstream.

The term scalability refers to the capability of a compressed sequence to be decoded at different data rates. In other words, a scalable multimedia clip can be edited relatively easily while it is compressed so that it can be streamed over channels with different bandwidths and still be decoded and played back in real-time.

Scalable multimedia is typically ordered so that there are hierarchical layers of data. A base layer contains a basic representation of the multimedia clip whereas enhancement layers contain refinement data on top of underlying layers. Consequently, the enhancement layers improve the quality of the clip.

Scalability is a desirable property for heterogeneous and error prone environments. This property is desirable in order to counter limitations such as constraints on bit rate, display resolution, network throughput, and decoder complexity.

Scalability can be used to improve error resilience in a transport system where layered coding is combined with transport prioritisation. The term transport prioritisation here refers to various mechanisms to provide different qualities of service in transport, including unequal error protection, to provide different channels having different error/loss rates. Depending on their nature, data are assigned differently, for example, the base layer may be delivered through a channel with high degree of error protection, and the enhancement layers may be transmitted through more error-prone channels.

Generally, scalable multimedia coding suffers from a worse compression efficiency than non-scalable coding. In other words, a multimedia clip coded as a scalable multimedia clip with all enhancement layers requires greater bandwidth than if it had been coded as a non-scalable single-layer clip with equal quality. However, exceptions to this general rule exist, for example the temporally scalable B-frames in video compression.

In the following, scalability is discussed with reference to the ITU-T H.263 video compression standard. H.263 is an ITU-T recommendation for video coding in low bit rate communication which generally means data rates below 64 kbps. The recommendation specifies the bitstream syntax and the decoding of the bitstream. Currently, there are two versions of H.263. Version 1 consists of the core algorithm and four optional coding modes. H.263 version 2 is an extension of version 1 providing twelve new negotiable coding modes.

Pictures are coded as luminance and two colour difference (chrominance) components (Y, C_B and C_R). The chrominance pictures are sampled with a half of the pixels along the both co-ordinate axes when compared to the luminance picture.

The scalability mode (Annex O) of H.263 specifies syntax to support temporal, signal-to noise ratio (SNR), and spatial scalability capabilities.

Spatial scalability and SNR scalability are closely related, the only difference being the increased spatial resolution provided by spatial scalability. An example of SNR scalable pictures is shown in Figure 6. SNR scalability implies the creation of multi-rate bit streams. It allows for the recovery of coding errors, or differences between an original picture and its reconstruction. This is achieved by using a finer quantizer to encode the difference picture in an enhancement layer. This additional information increases the SNR of the overall reproduced picture.

Spatial scalability allows for the creation of multi-resolution bit streams to meet varying display requirements and/or constraints. A spatially scalable structure is illustrated in Figure 7. It is essentially the same as in SNR scalability except that a spatial enhancement layer attempts to recover the coding loss between an up-sampled version of the reconstructed reference layer picture and a higher resolution version of the original picture. For example, if the reference layer has a quarter common intermediate format (QCIF) resolution, and the enhancement layer has a common intermediate format (CIF) resolution, the reference layer picture must be scaled accordingly such that the enhancement layer picture can be predicted from it. The QCIF standard allows the resolution to be increased by a factor of two in the vertical direction only, horizontal direction only, or both the vertical and horizontal directions for a single enhancement layer. There can be multiple enhancement layers, each increasing the picture resolution over that of the previous layer. The interpolation filters used to up-sample the reference layer picture are explicitly defined in the H.263 standard. Aside from the up-sampling process from the reference to the enhancement layer, the processing and syntax of a spatially scaled picture are identical to those of an SNR scaled picture.

In either SNR or spatial scalability, the enhancement layer pictures are referred to as EI- or EP-pictures. If the enhancement layer picture is upwardly predicted from a picture in the reference layer, then the enhancement layer picture is referred to as an Enhancement-I (EI) picture. In this type of scalability, the reference layer means the layer "below" the current enhancement layer. In some cases, when reference layer pictures are poorly predicted, over-coding of static parts of the picture can occur in the enhancement layer, causing an unnecessarily excessive bit rate. To avoid this problem, forward prediction is permitted in the enhancement layer. A picture that can be predicted in the forward direction from a previous enhancement layer picture or, alternatively, upwardly predicted from the reference layer picture is referred to as an Enhancement-P (EP) picture. Note that computing the average of the upwardly and forwardly predicted pictures can provide bi-directional prediction for EP-pictures. For both EI- and EP-pictures, upward prediction from the reference layer picture implies that no motion vectors are

required. In the case of forward prediction for EP-pictures, motion vectors are required.

In multi-point and broadcast multimedia applications, constraints on network throughput may not be foreseen at the time of encoding. Thus, a scalable bitstream should be used. Figure 8 shows an IP multicasting arrangement where each router can strip the bitstream according to its capabilities. It shows a server S providing a bitstream to a number of clients C. The bitstreams are routed to the clients by routers R. In this example, the server is providing a clip which can be scaled to at least three bit rates, 120 kbit/s, 60 kbit/s and 28 kbit/s.

If the client and server are connected via a normal uni-cast connection, the server may try to adjust the bit rate of the transmitted multimedia clip according to the temporary channel throughput. One solution is to use a layered bit stream and to adapt to bandwidth changes by varying the number of transmitted enhancement layers.

According to a first aspect of the invention there is provided a method of transmitting a data signal having a sequence of data units in a predetermined order over a transmission link between a data source and a data sink comprising the step of sending the data units in an order determined by their relative importance rather than their predetermined order.

According to a second aspect of the invention there is provided a data transmission system for transmitting a data signal having a sequence of data units in a predetermined order over a transmission link between a data source and a data sink the data units being sent in an order determined by their relative importance rather than their predetermined order.

According to a third aspect of the invention there is provided a server for transmitting a data signal having a sequence of data units in a predetermined

order over a transmission link the data units being sent in an order determined by their relative importance rather than their predetermined order.

Preferably the server comprises re-ordering means. Preferably the server comprises an editor for providing a scalable data signal.

According to a fourth aspect of the invention there is provided a computer program product stored on a computer usable medium comprising computer readable program means for causing transmission of a data signal having a sequence of data units in a predetermined order over a transmission link the data units being sent in an order determined by their relative importance rather than their predetermined order.

Preferably the computer program product comprises a server. In this sense it is meant that the when the computer program is loaded into hardware it causes the hardware to operate as a server device. Preferably the computer program product comprises an editor for providing a scalable data signal. Preferably the computer program product comprises re-ordering means for re-ordering the data units.

According to a fifth aspect of the invention there is provided a data signal having a sequence of data units for transmission over a transmission link between a data source and a data sink the data units being in an order determined by their relative importance rather than their predetermined order.

Preferably where the data units or layers of the data units are ordered into a sequence other than their original sequence, the invention also includes means or a step for returning the data units or layers of the data units into their original sequence.

The transmission link may simply be a bottleneck where congestion is occurring and the invention is used to overcome the bottleneck.

An advantage of the invention is that it can provide larger safety times than prior art arrangements with the same initial buffering. In addition, it provides some degree of protection against decreased transmission throughput. Therefore it helps in reducing the occurrence of pauses without sacrificing compression efficiency.

The invention also helps in transmitting over a unreliable network connection using transmission protocols which do not automatically retransmit but only retransmit those missing parts chosen by a higher level function, for example in response to feedback. In this case, it provides more time for retransmission of the vital parts of a data signal than prior art arrangements. Thus, vital parts of a stream are more likely to be transmitted successfully and the ultimate end user of the data signal is likely to perceive a better output quality.

Preferably the bandwidth of the transmission link available to the data signal is variable. This may be because the amount of data traffic using the link is variable, for example due to the need to re-transmit lost or corrupted data.

The invention provides a way of altering a data signal so that it can be transmitted over a transmission link having a bandwidth which would otherwise not be sufficient for it or for allowing certain of the bandwidth of its transmission link to be made available for other data, for example control information.

Preferably the effect of the invention is to increase the safety times of some data units relative to, or even at the expense of, the safety times of other data units.

Preferably the data signal is arranged in a hierarchy having a base layer and at least one enhancement layer. The term base layer means a most fundamental layer. The base layer may represent the least refined version of the data signal. In each unit, the (or each) enhancement layer, when combined with the base layer, refines the base layer to make it more like the data unit. Each enhancement layer

may add more refinement to the base layer. Preferably the safety time of the base layer is increased relative to the safety times of the enhancement layers.

Preferably the data signal is scalable. The base layer and the or each enhancement layer may have a scaled relationship with respect to each other. The signal may be scalable in the temporal, the spatial, the spectral or the SNR domains.

Preferably by virtue of re-ordering the sequence the layers have safety times which depend on their position in the hierarchy. For example the base layer of a particular data unit may have a greater safety time than enhancement layers. Advantageously, the finer the level of refinement represented by a layer, the shorter its safety time.

Preferably the source is a server. Alternatively it is an editor.

Preferably the sink is a client. It may be a destination of the data signal such as a mobile terminal, for example a mobile telephone.

Preferably the data units are frames. In one embodiment the data units represent a sequence of pictures. Conveniently it is a video sequence. Most preferably, it comprises multimedia data.

It should be understood that the terms "layer" and "layers" are being used figuratively and refer to blocks of data which can be applied to each other rather than to a strictly positional relationship.

The invention will now be described by way of example only with reference to the accompanying drawings in which:

Figure 1 shows a multimedia content creation and retrieval system;

Figure 2 shows operations of a multimedia clip editor;

Figure 3 shows operations of a multimedia server;

Figure 4 shows operations of a multimedia retrieval client;
Figure 5 shows an illustration of B picture prediction;
Figure 6 shows an illustration of SNR scalability;
Figure 7 shows an illustration of spatial scalability;
Figure 8 shows scalable multimedia being used in an IP multicast system;
Figure 9 shows mobile multimedia retrieval system according to the invention;
Figure 10 shows operations of an editor according to the invention;
Figure 11 shows operations of a client according to the invention;
Figure 12 shows safety times of a re-ordered sequence and an original sequence;
Figure 13 shows receiving and playback of a re-ordered sequence;
Figure 14 also shows receiving and playback of a re-ordered sequence; and
Figure 15 shows another multimedia content creation and retrieval system.

Figures 1 to 8 have been discussed above.

A mobile multimedia retrieval system 10 according to the invention is shown in Figure 9. Although only a single client 22 is shown it will be appreciated that there can be, and in preferred embodiments will be, a number of such clients. It will also be appreciated that at least part of the transmission link connecting a server 18 and the client 22 will be wireless.

The system 10 comprises an editor 12 which produces a compressed clip, a server database 14 for storing the compressed clip, re-ordering means 16 for re-ordering the sequence of the compressed clip, a server 18 for sending the re-ordered compressed clip and a client comprising order recovery means 20 to recover the original sequence of the compressed clip. The client 22 controls the operation of the server, for example by specifying which data are sent and how they are sent. Clearly, the re-ordering means 16 and the order recovery means 20 need to be compatible so that the latter can recover the original order. The re-ordering means 16 could be located to operate on data before they reach the server database 14 rather than the embodiment shown in Figure 9. Certain types of data, such as a live stream, may by-pass the database 14 and be sent by the

editor 12 directly to the re-ordering means 16. Although they are shown separately in Figure 9, the editor 12, the server database 14, the re-ordering means 16 and the server 18 can all be configured into a single unit or entity which can also be referred to as a server.

Figure 10 shows operation of an editor according to the invention. Raw data captured in the form of one or more uncompressed streams are edited to produce one or more uncompressed tracks. These tracks are compressed in a compression step to produce compressed tracks. The compressed tracks are multiplexed together to produce a compressed multimedia clip. The multimedia clips are coded scalably. The resulting bitstream consists of basic units, such as frames in video compression. These basic units are not all of equal importance; some are primary basic units which are vital to decode the bitstream and some are secondary basic units which improve the quality of the bitstream. Elimination of the secondary basic units causes only a quality loss in the decoded multimedia clip. In effect, the primary basic units are a base layer of the bitstream and the secondary units are enhancement layers.

Instead of organising the basic units into the original transmission order, that is the sequence in which they are produced by editing and compression of raw data, the invented method re-orders their sequence by prioritising the primary basic units so that they pre-empt the secondary basic units in the transmission order. Reordering can take place before or after the multiplexing process depending on the nature of the system and the data. In Figure 10 it is shown as taking place after multiplexing.

A re-ordering algorithm is used to re-order the sequence. In a preferred embodiment, the algorithm conducts stream analysis to identify primary and secondary basic units and to collect bit rate statistics. These may include the number of bits per frame. The bitstream is then re-ordered with iterative re-ordering steps to minimise the initial buffering time needed for full-quality playback with full channel throughput and reduced-quality playback with a reduced channel throughput. An example of such a re-ordering algorithm is given below. It should

be noted that the expected reduced channel throughput has to be estimated in advance since it is one of the input parameters of the algorithm. The estimation can be based on the statistics obtained from the channel behaviour, for example.

Re-ordering of the basic units means that the primary basic units are likely to be received earlier than indicated in their playback time even though some decrease in the channel throughput occurs. However, secondary basic units are vulnerable to transmission delay variations since the method schedules them to be received just before their indicated playback time. In this way the client is still likely to be able to play the primary basic units without pausing the playback, even if the channel throughput decreases considerably. The secondary basic units, on the other hand, may arrive later than their scheduled playback time and will not be used in playback.

Figure 11 shows operation of a client according to the invention. The bitstream is received by the client, re-ordered to produce the original compressed clip, demultiplexed to produce the compressed tracks and then decompressed to reconstruct the tracks. The basic unit order recovery step in the client can be applied either before or after demultiplexing (depending in when it occurs in the content editor). In Figure 11 it is shown as occurring before demultiplexing, to correspond to the order described in Figure 10. In this way re-ordering and order recovery are both applied to the multiplexed bitstream. Order recovery organises the basic units back into their natural sequence for decompression and discards all late-arriving secondary basic units.

On receipt of a basic unit by the client, it is time-stamped with a time by which it should be played. This time-stamp is compared with the current playback time. If the time-stamp is greater than the current playback time the basic unit is not decompressed and it is discarded. If the time-stamp is less than the current playback time the basic unit is decompressed and the two times are compared after decompression. If the time stamp is less than the current playback time the basic unit is played. Otherwise, it is discarded.

In the particular case in which re-ordering is performed after multiplexing, the multiplexing/demultiplexing format used needs to allow re-ordering of the basic units and recovery of the original order.

Figure 15 shows another view of a multimedia content creation and retrieval system according to the invention. The system 50 comprises an editor 52, a server 54 and a client 56.

The editor comprises a compressor 58 for compressing a raw data stream typically being in the form of a sequence of images into a form suitable for transmission, a multiplexer 60 for multiplexing the compressed raw data stream with other compressed data streams together to produce a compressed bitstream and a layer generation means 62 for producing primary basic frames which are vital to decode the bitstream and secondary basic frames which improve the quality of the bitstream.

The server 54 comprises re-ordering means 64 for arranging the frames in an order which depends on their importance in helping to reconstruct the raw data stream at the client 56 rather than their natural sequential order.

The server 54 and the client 56 each comprise transmitting/receiving means 66 and 70 respectively for data transmission over a transmission link 68 between the server 54 and the client 56.

The client 56 comprises order recovery means 72 for putting the frames back into their natural sequential order, a demultiplexer 74 for demultiplexing the bitstream and a decompressor 76 for decompressing the compressed data streams to reproduce the raw data stream. The client comprises playing/output means 78 for playing or otherwise outputting the raw data stream. The client also comprises a controller 80 which applies a control signal to the server 54 to control the re-ordering means in response to receipt of frames in the client 56.

Example

An embodiment of the invention was tested using an H.263 compliant encoder and a frame re-ordering tool. The encoder was capable of producing B-frames. The re-ordering tool comprised a re-ordering algorithm and could be adjusted to produce frame ordering for different decreased channel data rates.

A standard ITU-T test sequence of video frames called Glasgow was used. The sequence had a QCIF resolution, that is a luminance picture size of 176 x 144 pixels and a chrominance picture size of 88 x 72 pixels.

The first 297 frames of the Glasgow sequence were compressed to produce a temporally scalable bitstream comprising INTRA (I) frames, INTER (P) frames and bi-directionally predicted (B) frames. The I and P frames are referred to as primary frames and the B frames are referred to as secondary frames. The frame rate of the original (default) sequence was 12.5 frames per second (fps) and so the compressed sequence corresponded to 23.68 seconds. Each frame was compressed. The first frame was an INTRA frame, and the second one was an INTER frame. After that, there were alternate INTER frames and B-frames. A constant quantizer was used to produce a bitstream having an average bit rate of 41073 bps consisting of an average bit rate for the primary frames of 28984 bps and an average bit rate for the secondary frames of 12089 bps.

For the sake of simplicity, the example does not include multiplexing of the bitstream with another bitstream.

Re-ordering of the frame sequence was optimised separately for bit rates corresponding to sums of the primary frame bit rate (A) and (i) 25%; (ii) 50%; and (iii) 75 % of the secondary frame bit rate (B). It should be noted that 100% of the secondary bit rate (B) corresponds to the average bit rate required in order to obtain full quality playback. The summed bit rates were 32007, 35029 and 38051 bps respectively. For convenience, in the following description, these bit rates are

referred to as 25% (A+0.25B), 50% (A+0.5B) and 75% (A+0.75B), and the corresponding re-ordered sequences are referred to as 25-ordered, 50-ordered and 75-ordered. It was expected that the full rate of the channel would be the average bit rate of the sequence.

The reordering algorithm used two values:
FQFR (full quality full rate), which is the minimum initial buffering time to ensure full-quality playback assuming a full rate channel; and
RQRR (reduced quality reduced rate) which is the minimum initial buffering time to ensure reduced-quality playback in which the primary frames are played and some or all secondary frames may not be played (assuming a reduced secondary bit rate of 25%, 50% or 75%).

In calculating FQFR and RQRR for a particular channel, the algorithm assumes that it has a constant bit rate. Using the bit rate statistics the likely number of bits per frame is calculated. The time instant at which a particular frame is to be played is known from the time stamp. A time is determined by which a frame will be ready for display as a picture (allowing, of course, for the time needed to receive the bits for a frame and to prepare them) and this time is compared with the time stamp of the frame. If the frame will be received after it is to be displayed, the buffering time is increased, if it is received in time then the next frame can be dealt with. Once the algorithm has compared the likely times of receipt of all of the frames against their playback times and adjusted to buffering time accordingly so that they should all be received in time, the minimum buffering time has been calculated.

The algorithm was applied to a sequence having its original order. In this case, FQFR is smaller than RQRR because in order to obtain full quality playback, you have to wait longer in starting playback if you have a reduced rate channel. The algorithm seeks to minimise the initial buffering time needed for full-quality playback with full channel throughput and at the same time for reduced-quality playback with a reduced channel throughput. In other words, the algorithm seeks to find a value where FQFR and RQRR are approximately equal. It should be

noted that the algorithm is not optimal, that is the algorithm does not necessarily re-order the sequence so that the minimum initial buffering delay is found. The algorithm consists of the following steps:

1. Calculate FQFR and RQRR.

2. If FQFR is greater than or equal to RQRR, stop re-ordering. The minimum initial buffering delay is the greater of FQFR or RQRR. Otherwise, continue with the next step.

3. Shift all primary frames towards the beginning of the sequence by one frame step. For example, if the initial sequence is:

I0, P1, P2, B3, P4, B5, P6, B7, P8, B9, P10, ...,

shifting changes the sequence to:

I0, P1, P2, P4, B3, P6, B5, P8, B7, P10, B9, ...

where I0 is the first frame. It can be seen that where a secondary frame is earlier in sequence than an adjacent primary frame, the positions of the primary and secondary frames are exchanged.

4. Continue from step 1.

The buffering time needed to ensure pause-less playback in the reduced bit rate channel is greater for the original sequence than for the re-ordered sequence. It is desirable to have a short initial buffering time. The re-ordering process makes a trade-off between the minimum buffering time needed for a full-rate channel and the minimum buffering time needed for expected reduced-rate channels. Without re-ordering, less initial buffering time would have been needed to provide full-quality playback in the full-rate channel. However, without re-ordering, more initial buffering time would have been required to provide reduced-quality playback in the reduced-rate channel.

In the example, safety times were also investigated. Initial buffering time was set to be the minimum for reduced quality playback of the re-ordered sequence at 25% secondary bit rate. Figure 12 shows how safety time varies with playback of the 25-ordered sequence and the original order sequence. It shows that throughout playback, the 25-ordered sequence (represented by the upper line) has a longer safety time than the original order sequence.

Results were also obtained for the 50% and 75% secondary bit rates. The average safety times (in seconds) for all primary frames in a sequence are shown in the following table for all three reduced bit rates. The safety times are characterised by minimum safety time and average safety time.

		25%	50%	75%
Minimum	Re-ordered	4.79	2.51	1.06
	Default	3.46	1.63	0.64
	Difference	1.33	0.88	0.42
Average	Re-ordered	5.90	3.51	2.12
	Default	4.66	2.83	1.84
	Difference	1.24	0.67	0.28

It can be seen that both the minimum and average safety times of the re-ordered sequences are longer than those of the original sequence.

In contrast, set out below is a table showing the required buffering delays in seconds for full-quality playback at different average bit rates. This is to minimise pauses in playback.

	25%	50%	75%
Re-ordered	10.04	6.03	3.53
Default	7.02	4.64	3.02

It can be seen that the re-ordered sequence needs a longer buffering delay for full-quality playback because the secondary frames are transmitted out of their natural order and time is required to put all the frames back into their natural order. However, this is not important because the invention sets out to provide reduced quality playback with minimal initial buffering delay. To illustrate this, the following

table shows the buffering delays required to ensure pause-less reduced quality playback:

	25%	50%	75%
Re-ordered	5.28	3.49	2.47
Default	6.91	4.46	2.84

These results show that where the bit rate of the transmission channel is reduced, the invented method requires a shorter buffering delay before playback can be started.

The effect of the invention is illustrated in Figure 13 which shows playback of the 25-ordered sequence at 25% bit rate (that is A+0.25B as discussed above). Playback times, that is the times at which the frames are intended or supposed to be played to a viewer, are represented as a straight line. To be played in sequence, without requiring pauses, each frame must be received before its playback time. Any symbol located below the playback time line represents a frame which is received before its scheduled playback time and any symbol located above the playback time line represents a frame which is received after its scheduled playback time. In the example of Figure 13, the primary frames were all received in time to be played and the secondary frames were all received too late. Therefore the secondary frames were all discarded without being played.

It should be understood that if the sequence had not been re-ordered, there is a likelihood that some of the primary frames would have arrived after their playback times and so would have required playback to be paused (since a primary frame cannot be discarded without also discarding all of the frames predicted from it).

Figure 14 shows how the 25-ordered sequence is received and played when the channel throughput is not reduced. It can be seen that all frames are received in time to be played. Again, the upper line represents when secondary frames are received and the lower line represents when primary frames are received.

Of course, the exact appearance of Figures 13 and 14 depends on the nature of the original sequence, as well as the bit rate and the re-ordering that is performed. Differences can be expected in other systems which re-order data in different ways.

In use, the invention can be applied on-line to prepare a bitstream for transmission at some future point. For example, assuming that there may be considerable variation in the available bandwidth of a given transmission channel, the algorithm may be applied to a bitstream and re-ordered sequences may be calculated for several alternative bandwidths, such as 25%, 50% and 75%. In transmission, if problems are encountered, for example with pausing, the re-ordered sequence currently being used may be changed to one better suited to a lower available bandwidth. Equally, if transmission is proceeding better than expected, a sequence better suited for a higher bandwidth may be used. The invention may check the progress of transmission and/or playback a number of times during transmission and adjust the sequence accordingly. It is advantageous if this adjustment can be programmed dynamically whilst transmission is occurring. In this way, intermittent problems can de dealt with.

The part of the invention which re-orders a bitstream is located in an editor or in a server. The part of the invention which recovers the order is located in a client.

A further improvement is provided if the proportion of bits belonging to the primary frames is reduced by replacing some of the primary frames with B-frames. This reduces the compression efficiency slightly because B-frames are predicted from reference frames (P-frames) which are further away in time and so the B-frames and P-frames are less alike. This yields a poorer predicted B-frame and so more bits must be used to code the corresponding prediction error image. Since in this arrangement there are now more bits belonging to secondary frames which can be discarded, re-ordering the frames can provide even better protection against transmission delays and pauses.

One way of looking at the invention is to say that it sacrifices the safety times of the secondary frames in order to maintain maximum safety times for the primary frames.

The invention is particularly suitable for streaming-type multimedia retrieval.

While preferred embodiments of the invention have been shown and described, it will be understood that such embodiments are described by way of example only. For example, although the invention has been described applied to a temporally scalable bitstream, it can also be applied to other types of scalability, including spatial, spectral and SNR. Numerous variations, changes and substitutions will occur to those skilled in the art without departing from the scope of the present invention. Accordingly, it is intended that the following claims cover all such variations as fall within the spirit and scope of the invention.

Claims

Claims

1. A server for transmitting a data signal having a sequence of data units in a predetermined order over a transmission link the data units being sent in an order determined by their relative importance rather than their predetermined order.

2. A server according to claim 1 in which the data units represent a base layer and at least one enhancement layer.

3. A server according to claim 2 comprising re-ordering means to change the order of the data units.

4. A server according to any preceding claim in which the data signal is scalable.

5. A server according to claim 4 in which the signal is scalable in a domain selected from a group consisting of the temporal, the spatial, the spectral and the SNR domains.

6. A server according to any preceding claim which comprises an editor for providing the data signal.

7. A server according to any preceding claim in which the data signal represents a sequence of pictures to produce a moving image.

8. A server according to claim 7 in which the data signal represents a video sequence.

9. A server according to any preceding claim in which the data signal comprises multimedia data.

10. A data transmission system for transmitting a data signal having a sequence of data units in a predetermined order over a transmission link between a data

source and a data sink the data units being sent in an order determined by their relative importance rather than their predetermined order.

11. A transmission system according to claim 10 in which the data units each comprise a base layer and at least one enhancement layer and when re-ordered the base layer of a particular data unit has a greater safety time than of the or each enhancement layer of the particular data unit.

12. A transmission system according to claim 10 or claim 11 in which the source is a server.

13. A transmission system according to any of claims 10 to 12 in which the source is an editor.

14. A transmission system according to any of claims 10 to 13 in which the sink is a client.

15. A transmission system according to any of claims 10 to 14 in which the sink is a mobile terminal.

16. A transmission system according to claims 10 to 15 in which the sink is a mobile telephone.

17. A transmission system according to claims 10 to 16 in which means are provided to check the progress of transmission and to change the order being used to one better suited to available bandwidth.

18. A method of transmitting a data signal having a sequence of data units in a predetermined order over a transmission link between a data source and a data sink comprising the step of sending the data units in an order determined by their relative importance rather than their predetermined order.

19. A method according to claim 18 in which the data units are returned to their original sequence once they have been transmitted over the transmission link.

20. A method according to claim 18 or claim 19 in which the progress of transmission is checked and the order being used is changed to one better suited to available bandwidth.

21. A computer program product stored on a computer usable medium comprising computer readable program means for causing transmission of a data signal having a sequence of data units in a predetermined order over a transmission link the data units being sent in an order determined by their relative importance rather than their predetermined order.

22. A computer program product according to claim 21 comprising a server.

23. A computer program product according to claim 21 or claim 22 comprising an editor for providing a scalable data signal.

24. A computer program product according to any of claims 22 to 23 comprising re-ordering means for providing the layers of the or each data units with different safety times.

25. A data signal having a sequence of data units for transmission over a transmission link between a data source and a data sink the data units being in an order determined by their relative importance rather than their predetermined order.

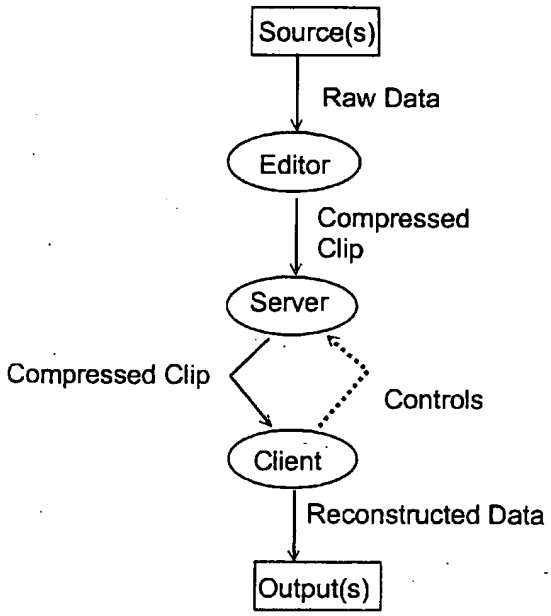

Source(s)
Raw Data
Editor
Compressed Clip
Server
Compressed Clip
Controls
Client
Reconstructed Data
Output(s)

Fig. 1

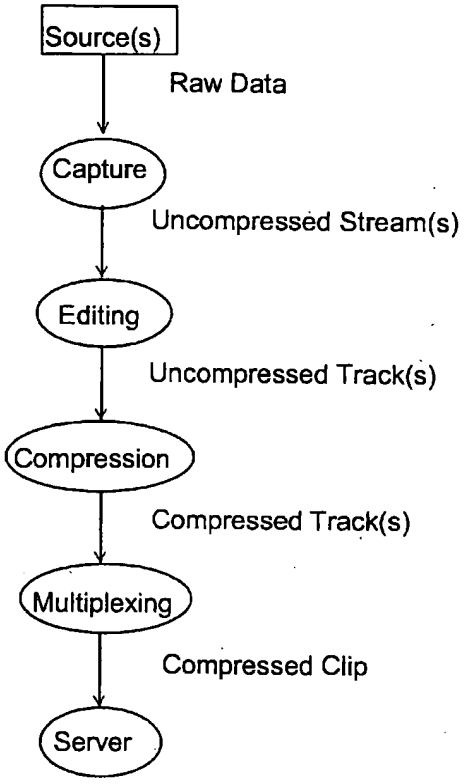

Source(s)
Raw Data
Capture
Uncompressed Stream(s)
Editing
Uncompressed Track(s)
Compression
Compressed Track(s)
Multiplexing
Compressed Clip
Server

Fig. 2

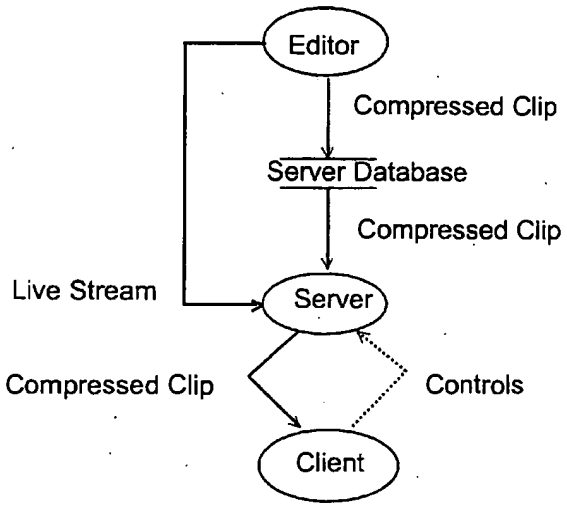

Editor
Compressed Clip
Server Database
Compressed Clip
Live Stream
Server
Compressed Clip
Controls
Client

Fig. 3

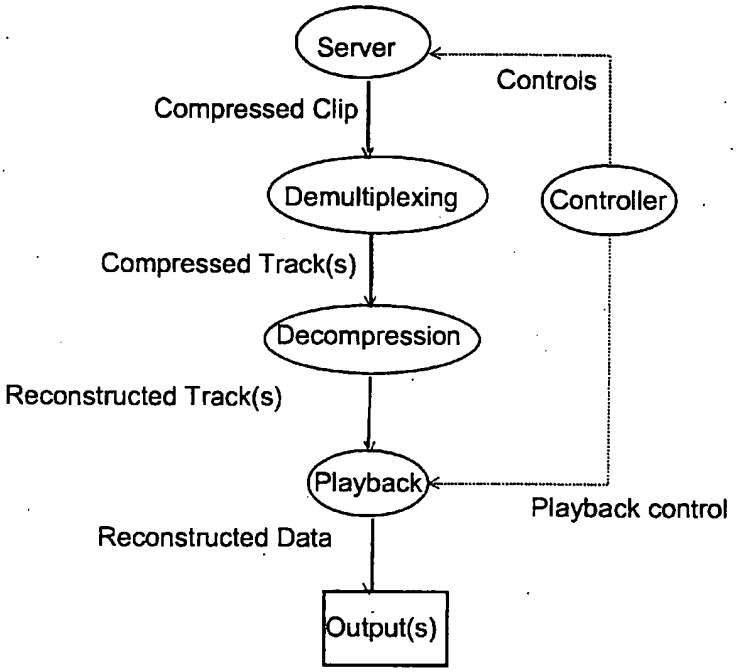

Server
Controls
Compressed Clip
Demultiplexing
Controller
Compressed Track(s)
Decompression
Reconstructed Track(s)
Playback
Playback control
Reconstructed Data
Output(s)

Fig. 4

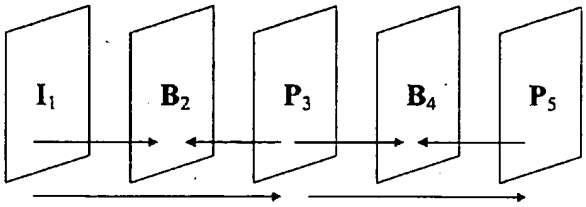


Fig. 5

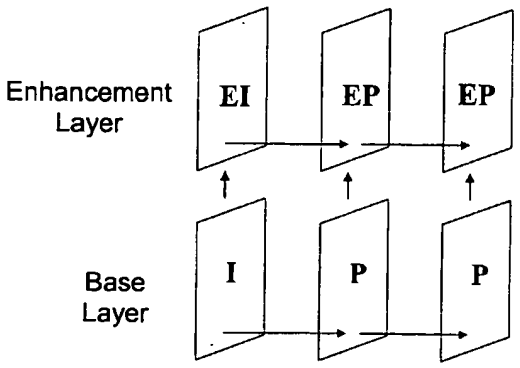


Fig. 6

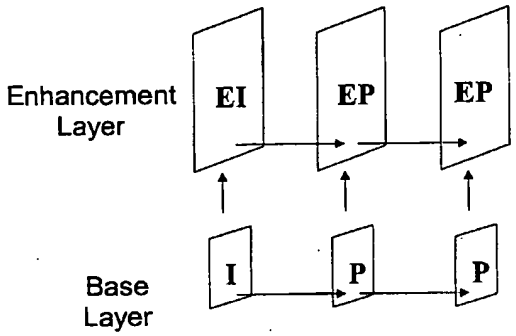

Enhancement Layer
EI
EP
EP
Base Layer
I
P
P

Fig. 7

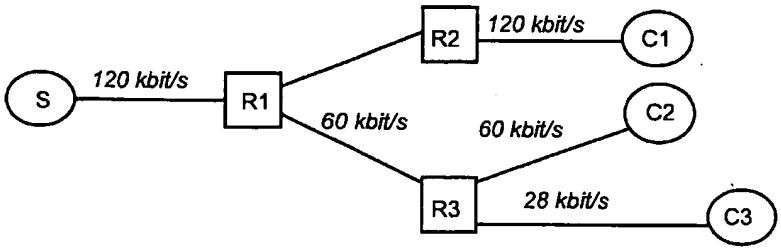

S
120 kbit/s
R1
R2
120 kbit/s
C1
60 kbit/s
60 kbit/s
C2
R3
28 kbit/s
C3

Fig. 8

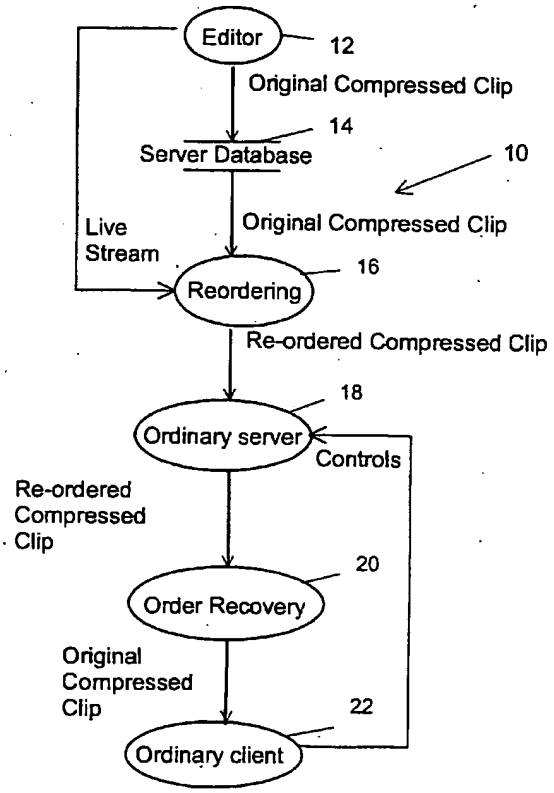

Editor
12
Original Compressed Clip
14
Server Database
10
Original Compressed Clip
Live
Stream
16
Reordering
Re-ordered Compressed Clip
18
Ordinary server
Controls
Re-ordered
Compressed
Clip
20
Order Recovery
Original
Compressed
Clip
22
Ordinary client

Fig. 9

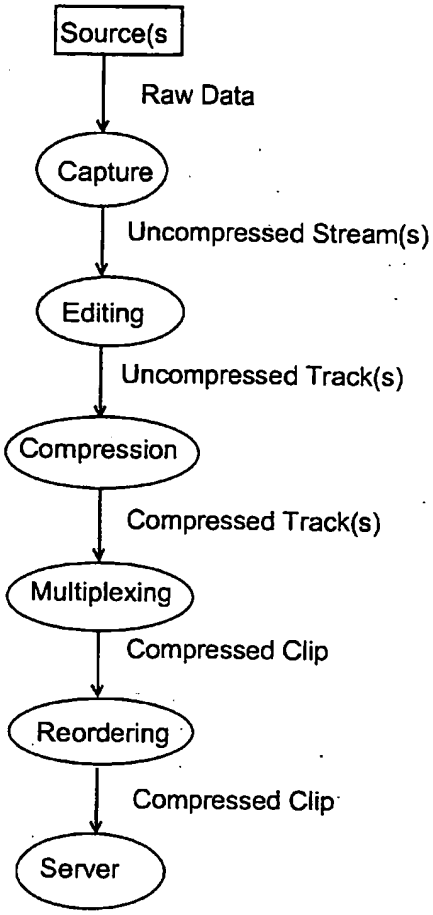

Source(s
Raw Data
Capture
Uncompressed Stream(s)
Editing
Uncompressed Track(s)
Compression
Compressed Track(s)
Multiplexing
Compressed Clip
Reordering
Compressed Clip
Server

Fig. 10

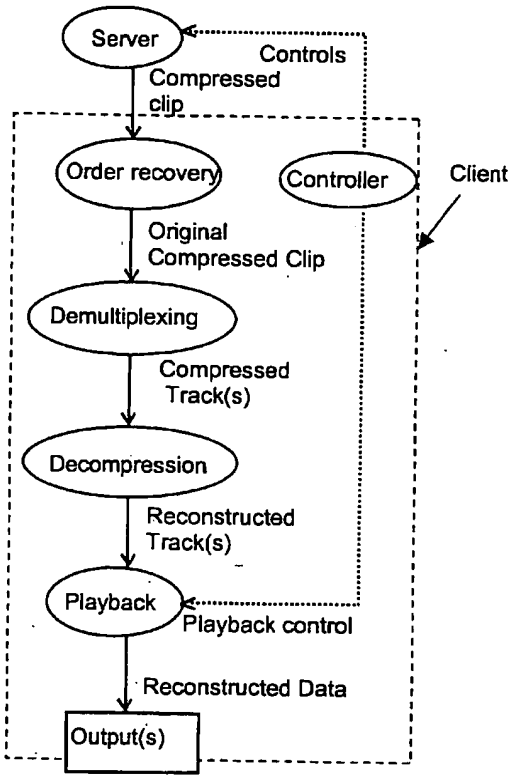

Server
Controls
Compressed clip
Order recovery
Controller
Client
Original Compressed Clip
Demultiplexing
Compressed Track(s)
Decompression
Reconstructed Track(s)
Playback
Playback control
Reconstructed Data
Output(s)

Fig. 11

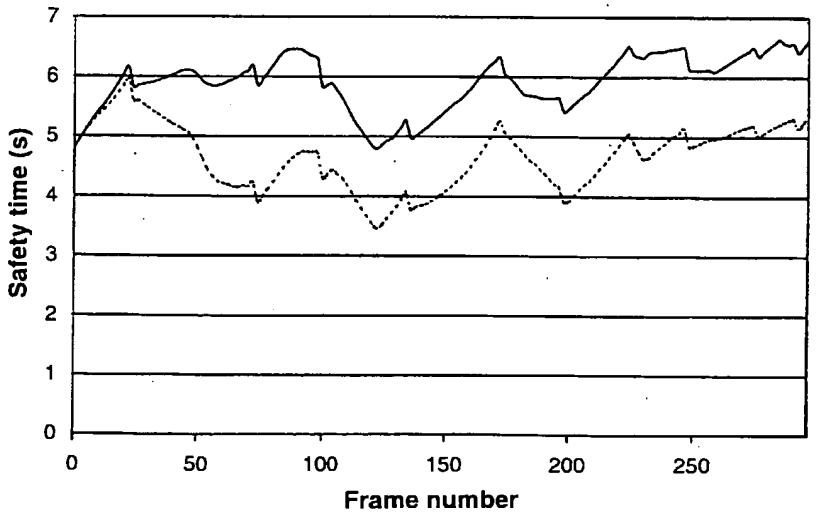


Fig. 12

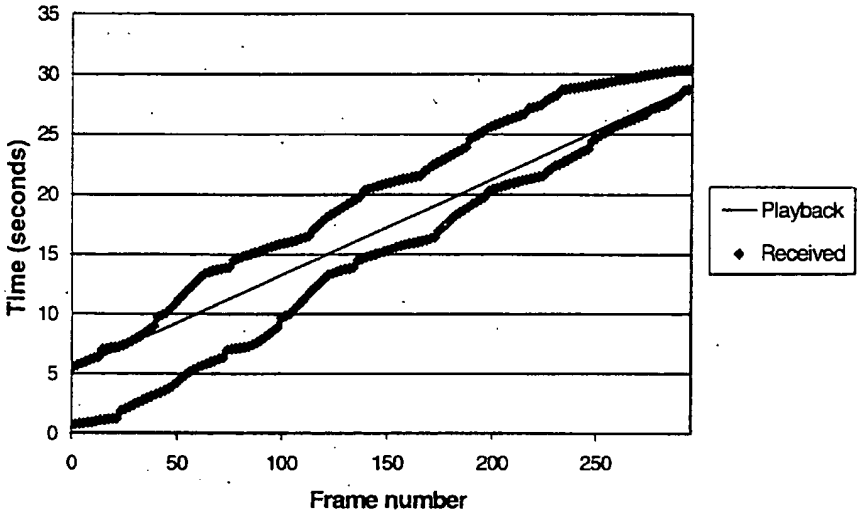

35
30
25
20
15
10
5
0
Time (seconds)
0
50
100
150
200
250
Frame number
Playback
Received

Fig. 13

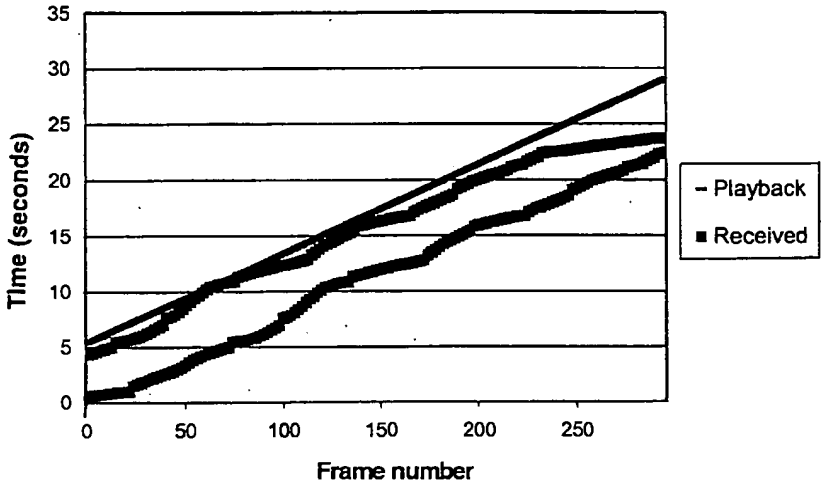

35
30
25
20
15
10
5
0
Time (seconds)
0
50
100
150
200
250
Frame number
Playback
Received

Fig. 14

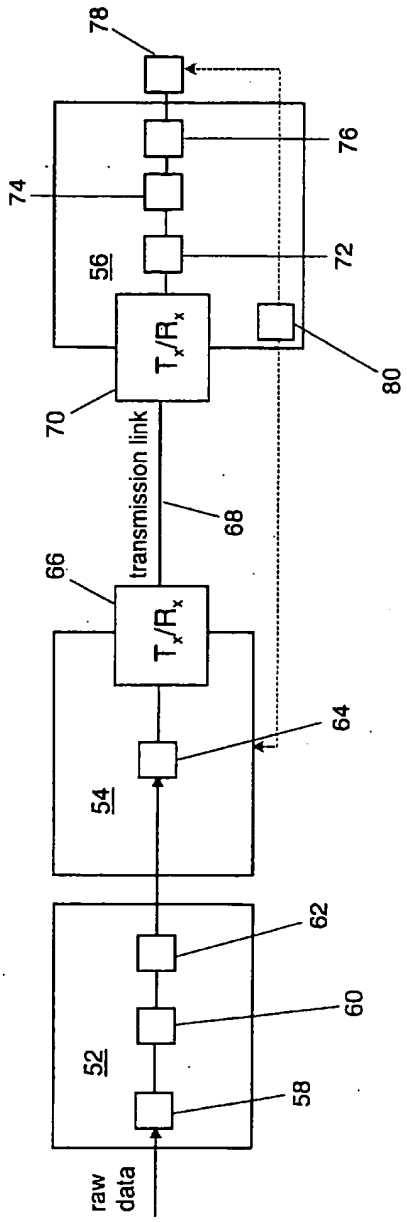

78
76
74
72
56
T_x/R_x
80
70
transmission link
68
66
T_x/R_x
64
54
62
60
52
58
raw data

Fig. 15

INTERNATIONAL SEARCH REPORT

Interna al Application No
PCT/FI 00/00363

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N5/00 H04N7/26

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
WPI Data, PAJ, EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PATENT ABSTRACTS OF JAPAN vol. 018, no. 566 (E-1622), 28 October 1994 (1994-10-28) & JP 06 205384 A (NEC CORP), 22 July 1994 (1994-07-22) abstract	1,6-10, 12-23,25
Y		2-5,11, 24

Y	US 5 564 001 A (LEWIS SCOTT W) 8 October 1996 (1996-10-08) column 2, line 58 -column 3, line 30 column 4, line 58 - line 65 column 7, line 4 - line 14 column 7, line 43 - line 47 abstract	2-5,11, 24
A		1,6-10, 12-23,25

	-/--	

[X] Further documents are listed in the continuation of box C. [X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
9 August 2000	- 9. 10. 2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Jesper Bergstrand

Form PCT/ISA/210 (second sheet) (July 1992)

page 1 of 2

INTERNATIONAL SEARCH REPORT

Interna al Application No
PCT/FI 00/00363

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 634 871 A (AT & T CORP) 18 January 1995 (1995-01-18) the whole document -----	1-25

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

...rmation on patent family members

Inter ...nal Application No
PCT/FI 00/00363

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
JP 06205384 A	22-07-1994	JP 2833950 B	09-12-1998
US 5564001 A	08-10-1996	US 5325423 A	28-06-1994
		AU 693426 B	02-07-1998
		AU 5605394 A	08-06-1994
		AU 8787298 A	03-12-1998
		BR 9307442 A	01-06-1999
		CA 2148090 A	26-05-1994
		EP 0746937 A	11-12-1996
		JP 8503585 T	16-04-1996
		NZ 258406 A	24-11-1997
		RU 2110162 C	27-04-1998
		WO 9411981 A	26-05-1994
EP 0634871 A	18-01-1995	CA 2126467 A	14-01-1995
		JP 7170514 A	04-07-1995
		US 5742343 A	21-04-1998

Form PCT/ISA/210 (patent family annex) (July 1992)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 June 2001 (28.06.2001)

PCT

(10) International Publication Number
WO 01/47283 A1

(51) International Patent Classification[7]: H04N 7/26

(21) International Application Number: PCT/US00/09584

(22) International Filing Date: 10 April 2000 (10.04.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/171,774 22 December 1999 (22.12.1999) US

(71) Applicant: **GENERAL INSTRUMENT CORPORATION** [US/US]; 101 Tournament Drive, Horsham, PA 19044 (US).

(72) Inventors: **WAN, Wade, K.**; Apartment 501, 12-16 Ellery Street, Cambridge, MA 02138 (US). **CHEN, Xuemin**; 8560 Foxcroft Place, San Diego, CA 92129 (US). **LUTHRA, Ajay**; 13695 Winstanley Way, San Diego, CA 92130 (US).

(74) Agent: **LIPSITZ, Barry, R.**; Building No. 8, 755 Main Street, Monroe, CT 06468 (US).

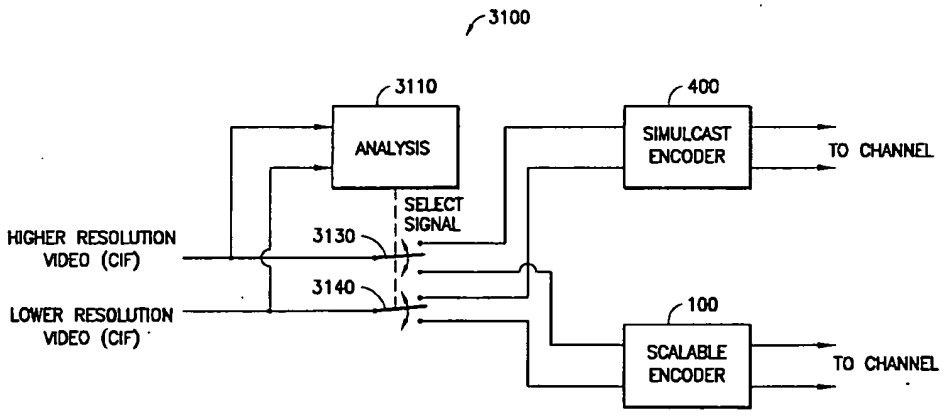
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: VIDEO COMPRESSION FOR MULTICAST ENVIRONMENTS USING SPATIAL SCALABILITY AND SIMULCAST CODING



(57) Abstract: An optimal system for determining whether simulcast coding (400) or spatial scalability coding (100) should be used to encode video for clients with a specific communication link. Operating points (A', B', C') for both simulcast coding and spatial scalability are also determined. Adaptive switching (3130, 3140) is provided, with the operating points and decision boundaries being used to guide the switching to optimize the quality of the higher-resolution data based on a bit allocation to the associated lower-resolution data. A system for determining the point (A', C') of equal quality in both layers of simulcast and spatial scalability coding is also disclosed. The proportion of bits allocated to the base layer to achieve equal quality is essentially independent of the total bit rate for both simulcast and spatial scalability.

WO 01/47283 A1

VIDEO COMPRESSION FOR MULTICAST ENVIRONMENTS USING SPATIAL SCALABILITY AND SIMULCAST CODING

BACKGROUND OF THE INVENTION

This application claims the benefit of U.S. Provisional Application No. 60/171,774, filed December 22, 1999.

The present invention relates to digital television and the like, and more particularly to a video coding scheme for multicast applications. The invention is particularly suitable for providing a streaming video server for multicast video over computer networks, such as Internet protocol (IP) networks. A multicast transmission can use simulcast ("Sim.") or spatial scalability ("SS") coding.

Usually, three major factors in a multicast video service need to be considered:

(1) The costs of reaching the audience (from the video-coding point of view);

(2) Quality of Service (e.g. visual quality); and

(3) Encoding complexity vs. decoding complexity.

Some multicast systems have chosen to use simulcast coding (the independent coding of bitstreams to achieve video scalability) exclusively. The simulcast approach does not require additional encoder or decoder complexity and thus satisfies the third factor listed above.

For some application environments, codec complexity may not be an issue and the first two factors are the main concerns. For these services, the selection of the video compression (coding) scheme to be used often follows the rule that for a given total combined allocation of two or more service rates, a coding scheme that can achieve the highest peak signal-to-noise ratio (PSNR) for clients is desired. Also, for a given quality requirement (e.g., PSNR for each bitstream), a coding scheme that can achieve the smallest total bit allocation is desired.

Intuitively, scalable bitstreams (dependently coded bitstreams) are expected to perform better than simulcast coding. This reasoning suggests that a multicast implementation without complexity constraints should only use spatial scalability. However, the present inventors have found that simulcast coding outperforms spatial scalability for certain operating regions. This surprising discovery enables a determination to be made as to which approach to use for a given application and to provide an adaptive switching technique between the two coding approaches.

It would be advantageous to provide an optimal method for selecting between spatial scalability and simulcast coding for multicast video services. Such a method should enable a determination to be made as to whether simulcast coding or spatial scalability should be used to encode video for clients with a specific

communication link. It would be further advantageous if, in addition to guiding decisions for specific cases, the methodology could be used to construct decision regions to guide more general scenarios, or used to adaptively switch between the two approaches. Operating points for both simulcast coding and spatial scalability, in terms of bit allocations among clients, should also be determinable by using such a method.

The present invention provides a system having the aforementioned and other advantages.

SUMMARY OF THE INVENTION

An optimal technique is provided for selecting between spatial scalability and simulcast coding to provide an efficient compression algorithm. In particular, simulcast coding can outperform spatial scalability when a small proportion of bits is allocated to the base layer.

A technique is also provided for determining whether simulcast coding or spatial scalability should be used to encode video for clients with a specific communication link. Operating points for both simulcast coding and spatial scalability are also determined. Adaptive switching between the two approaches is also provided, with the operating regions being used to guide the switching.

The invention also provides a method for determining the point of equal quality in both layers of simulcast coding. The proportion of bits allocated to the base layer to achieve equal quality is independent of the total bit rate for both simulcast and spatial scalability.

Corresponding methods and apparatuses are presented.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a general scalable coder/decoder (CODEC) with two layers;

FIG. 2 illustrates temporal scalability with two layers;

FIG. 3 illustrates spatial scalability with two layers;

FIG. 4 is a block diagram of a system for providing simulcast coding with two bitstreams;

FIG. 5 is a plot of PSNR vs. bit rate for the single layer coding of the QCIF carphone video sequence;

FIG. 6 is a plot of PSNR vs. bit rate for the single layer coding of the CIF carphone video sequence;

FIG. 7 is a plot of PSNR vs. the fraction of total bits allocated to the lower-resolution stream, for QCIF/CIF simulcast of the carphone video sequence for total bit rates of 0.29, 0.32, and 0.35 Mbps;

FIG. 8 is a plot illustrating an example of the iterations needed to obtain the point of equal quality in both layers of simulcast coding for the QCIF/CIF carphone video sequence.

FIG. 9 is a plot of PSNR vs. total bit rate for a QCIF/CIF simulcast, where the lower-resolution stream and higher-resolution stream have the same PSNR for a given total bitrate;

FIG. 10 is a plot of the fraction of total bits

allocated to the lower-resolution stream vs. the total bit rate for a QCIF/CIF simulcast, where the lower-resolution stream and higher-resolution stream have the same PSNR for a given total bitrate;

FIG. 11 is a plot of PSNR vs. total bit rate for QCIF/CIF spatial scalable coding of the carphone video sequence;

FIG. 12 is a plot of PSNR vs. the fraction of total bits allocated to the base layer, for QCIF/CIF spatial scalable coding of the carphone video sequence for total bit rates of 0.29, 0.32 and 0.35 Mbps;

FIG. 13 is a plot of PSNR vs. total bit rate for QCIF/CIF spatial scalability, where the base layer and the enhancement layer for a given video sequence have the same PSNR for a given total bitrate;

FIG. 14 is a plot of the fraction of total bits allocated to the base layer vs. total bit rate for QCIF/CIF spatial scalability, where the base layer and the enhancement layer of a given video sequence have the same PSNR for a given total bitrate;

FIG. 15 is a plot of PSNR vs. total bit rate for QCIF/CIF simulcast and spatial scalable coding of the carphone video sequence, where the QCIF and CIF video sequences have the same PSNR for a given total bitrate;

FIG. 16 is a plot of PSNR vs. total bit rate for QCIF/CIF simulcast and spatial scalable coding of the news video sequence, where the QCIF and CIF video sequences have the same PSNR for a given total bitrate;

FIG. 17 is a plot of PSNR vs. total bit rate for QCIF/CIF simulcast and spatial scalable coding of the foreman video sequence, where the QCIF and CIF video sequences have the same PSNR for a given total bitrate;

FIG. 18 is a plot of PSNR vs. total bit rate for QCIF/CIF simulcast and spatial scalable coding of the basket video sequence, where the QCIF and CIF video sequences have the same PSNR for a given total bitrate;

FIG. 19 is a plot of PSNR vs. total bit rate for QCIF/CIF simulcast and spatial scalable coding of the silentvoice video sequence, where the QCIF and CIF video sequences have the same PSNR for a given total bitrate;

FIG. 20 is a plot of PSNR vs. total bit rate for QCIF/CIF simulcast and spatial scalable coding of the bus video sequence, where the QCIF and CIF video sequences have the same PSNR for a given total bitrate;

FIG. 21 is a plot of PSNR vs. the fraction of total bits allocated to the base layer (for SS) or to the lower-resolution stream (for Sim.), for the carphone video sequence and a total bandwidth of 0.29 Mbps, which also illustrates the tradeoff between base and enhancement layers (for SS), and between the lower and higher resolution streams (for Sim.);

FIG. 22 is a plot of PSNR vs. the fraction of total bits allocated to the base layer (for SS) or to the lower-resolution stream (for Sim.), for the carphone video sequence and a total bandwidth of 0.32

Mbps;

FIG. 23 is a plot of PSNR vs. the fraction of total bits allocated to the base layer (for SS) or to the lower-resolution stream (for Sim.), for the carphone video sequence and a total bandwidth of 0.35 Mbps;

FIG. 24 is a plot of PSNR vs. the bit rate for the enhancement layer (for SS) or the higher-resolution stream (for Sim.) for the carphone video sequence and a fixed bit rate of 0.29 Mbps for the base layer (for SS) or the lower-resolution stream (for Sim.);

FIG. 25 is a plot of PSNR vs. the bit rate for the enhancement layer (for SS) or the higher-resolution stream (for Sim.), for the carphone video sequence and a fixed bit rate of 0.05 Mbps for the base layer (for SS) or the lower-resolution stream (for Sim.);

FIG. 26 is a plot of the bit rate for the enhancement layer (for SS) or the higher-resolution stream (for Sim.) vs. the bit rate for the base layer (for SS) or the lower-resolution stream (for Sim.), for simulcast and spatial scalability decision boundaries;

FIG. 27 is a plot of normalized bit rate for the enhancement layer (for SS) or the higher-resolution stream (for Sim.) vs. normalized bit rate for the base layer (for SS) or the lower-resolution stream (for Sim.), for simulcast and spatial scalability decision boundaries;

FIG. 28 is a block diagram illustrating an example

of multicast broadcasting using simulcast coding;

FIG. 29 is a block diagram illustrating an example of multicast broadcasting using spatial scalability coding;

FIG. 30 is a plot of bit rate for the enhancement layer (for SS) or the higher-resolution stream (for Sim.) vs. bit rate for the base layer (for SS) or the lower-resolution stream (for Sim.), for simulcast and spatial scalability decision regions; and

FIG. 31 illustrates an adaptive simulcast/spatial scalability encoder apparatus in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides techniques for selecting between simulcast coding and spatial scalability for multicast services, such as multicast video.

A brief review of general scalable coding (temporal scalability in addition to spatial scalability) is first provided. Simulations have been performed with an MPEG-4 codec to gain insight into the issues involved in transmitting the same video sequence at multiple spatial resolutions. In addition to deciding between simulcast coding and spatial scalability, one issue is the proper allocation of bits to the different layers.

General guidelines are provided and a scenario for achieving equal quality in both layers is examined in detail. The results obtained using quarter common intermediate format (QCIF) and common intermediate format (CIF) resolution sequences may be directly applied to applications such as video transmission in a multicast environment.

While QCIF is used as an example of lower-resolution data, and CIF is used as an example of higher-resolution data, these are examples only, and other suitable data formats can be used.

CIF was developed so that computerized video images can be shared from one computer to another. An

image that is digitized to CIF has a resolution of 352 x 288 or 352 x 240, which is essentially one-half of the resolution of CCIR 601. The CCIR 601 recommendation of the International Radio Consultative Committee for the digitization of color video signals deals with color space conversion from RGB to YCrCb, the digital filters used for limiting the bandwidth, the sample rate (defined as 13.5 MHz), and the horizontal resolution (720 active pixels).

Many applications desire the capability to transmit and receive video at a variety of resolutions and/or qualities. One method to achieve this is with scalable or layered coding, which is the process of encoding video into an independent base layer and one or more dependent enhancement layers. This allows some decoders to decode the base layer to receive basic video and other decoders to decode enhancement layers in addition to the base layer to achieve higher temporal resolution, spatial resolution, and/or video quality.

The general concept of scalability is illustrated in FIG. 1 for a codec with two layers. Note that additional layers can be used. The scalable encoder 100 takes two input sequences and generates two bitstreams for multiplexing at a mux 140. Specifically, the input base video stream or layer is processed at a base layer encoder 110, and upsampled at a midprocessor 120 to provide a reference image for

predictive coding of the input enhanced video stream or layer at an enhancement layer encoder 130.

Note that coding and decoding of the base layer operate exactly as in the non-scalable, single layer case. In addition to the input enhanced video, the enhancement layer encoder uses information about the base layer provided by the midprocessor to efficiently code the enhancement layer. After communication across a channel, which can be, e.g., a computer network such as the Internet, or a broadband communication channel such as a cable television network, the total bitstream is demultiplexed at a demux 150, and the scalable decoder 160 simply inverts the operations of the scalable encoder 100 using a base layer decoder 170, a midprocessor 180, and an enhancement layer decoder 190.

The MPEG-2 standard defines scalable tools for spatial, temporal and quality (SNR) scalability. The main commercial applications that MPEG-2 was targeted to were digital video disks and digital television, applications where the additional functionality of scalability is often not used. Thus, there has been limited commercial interest in MPEG-2 scalable coding in the past. However, new applications such as streaming video could greatly benefit from scalability. One example where scalable coding may be useful is for video transmission in a multicast environment. Clients have a wide range of processing power, memory resources and available bandwidth. This requires a server to

provide different resolutions and/or qualities of video to be able to satisfy the different capabilities of their clients. The recently completed multimedia standard MPEG-4 version 1 offers two types of scalability: temporal and spatial. In addition to applying scalability to frames (pictures) of video, the standard also defines scalability for arbitrary shaped objects.

This document focuses on frame-based scalability, although the concepts of the invention are generally applicable to arbitrarily shaped objects. In addition to temporal and spatial scalability, a third type of scalable coding for quality scalability called Fine Granular Scalability (FGS) is currently being evaluated for inclusion in MPEG-4. A brief review of temporal and spatial scalability in MPEG-4 is presented before discussion of simulcast coding.

Temporal scalability permits an increase in the temporal resolution by using one or more enhancement layers in addition to the base layer.

FIG. 2 shows an example of temporal scalable coding with two layers. Basic video is obtained by decoding only the independent base layer 200, which is done in the same manner as in the non-scalable, single layer case. Use of the dependent enhancement layer 250 provides video with, e.g., seven times the temporal resolution of the basic video. The same spatial resolution is obtained whether or not the enhancement

layer 250 is used. A frame in the enhancement layer 250 can use motion compensated prediction from the previous or next frame in display order belonging to the base layer as well as the most recently decoded frame in the same layer.

Spatial scalability permits an increase in the spatial resolution by using enhancement layers in addition to the base layer. FIG. 3 shows an example of spatial scalable coding with two layers. Basic video is obtained by decoding only the independent base layer 300, which is done in the same manner as in the non-scalable, single layer case. Use of the dependent enhancement layer 350 provides video with, e.g., twice the spatial resolution of the basic video. The same temporal resolution is obtained whether or not the enhancement layer is used. A frame in the enhancement layer can use motion compensated prediction from the temporally coincident frame in the base layer as well as the most recently decoded frame in the same layer.

Another method to transmit video at multiple resolutions or qualities is simulcast coding. FIG. 4 shows an example of simulcast coding with two bitstreams. For simulcast coding, the streams are independent, whereas scalable coding usually refers to an independent base layer with one or more dependently-coded enhancement layers. For comparison with scalable coding, one of the simulcast streams (termed a lower-resolution stream) has the same resolution as the base

layer, and the other simulcast bitstream (termed a higher-resolution stream) has the same resolution as the enhancement layer.

First and second input bitstreams are coded at corresponding video encoders 420 and 410, respectively. The input video #1 is assumed to be a lower-resolution stream, while the input video #2 is assumed to be a higher-resolution stream. This involves coding each representation independently and is usually less efficient than scalable coding since similar information in another bitstream is not exploited. The bitstreams are then multiplexed at a mux 430, transmitted across some channel, demultiplexed at a demux 440, and decoded independently at video decoders 470 and 460, respectively, in a simulcast decoder 450. Unlike scalable coding, no additional decoder complexity is required to decode the higher-resolution video. This may be important for commercial applications since additional decoder complexity often increases the cost of receivers.

This invention focuses on the performance of spatial scalability and its simulcast counterpart. Simulations have been performed with an MPEG-4 encoder on rectangular video to gain insight into the issues with transmitting video at a variety of spatial resolutions. One issue with layered coding is the proper allocation of bits between layers. In addition to examining the differences between simulcast coding

and spatial scalability, investigations in connection with the invention focus on determining guidelines for bit allocation.

Six different video sequences were examined, namely Basket, Bus, Carphone, Foreman, News, and Silentvoice. This set of sequences has a wide range of complexity, so the results should be generally applicable to other sequences. The Basket and Bus sequences have a large amount of motion and may stress most encoders. The News and Silentvoice sequences have large stationary backgrounds making them easy to compress efficiently. The original sequences were in CIF format (288 x 352 pixels) and QCIF format (144 x 176 pixels). The sequences were created by downsampling (without use of any anti-aliasing filter, i.e., decimation). Each sequence was 150 frames long and the source material and display frame rates were 30 frames per second. An MPEG-4 encoder was used to encode the simulcast and spatial scalable streams at various fixed quantization levels with no rate control. The parameters used for the simulations are shown in Table 1. "VOP" refers to a Video Object Plane, as known from the MPEG-4 standard.

	Lower Layer of Simulcast Streams	Upper Layer of Simulcast Streams	Base Layer of Scalable Streams	Enhancement Layer of Scalable Streams
Q_I	{4, 6, 8, …, 24}	{4, 6, 8, …, 24}	{4, 6, 8, …, 24}	{6, 8, 10, …, 24}
Q_P	min (1.4Q_I , 32)	min (1.4Q_I , 32)	min (1.4Q_I , 32)	min (1.4Q_I , 32)
Q_B	min (1.8Q_I , 32)	min (1.8Q_I , 32)	min (1.8Q_I , 32)	min (1.8Q_I , 32)
Structure	IPPBPPB…	IPPBPPB…	IPPBPPB…	PBBPBBPBB…
M	3	3	3	12
N	12	12	12	---
Range	8	16	8	16
Q_I	Quantizer for I-VOPs			
Q_P	Quantizer for P-VOPs			
Q_B	Quantizer for B-VOPs			
Structure	Picture Structure			
M	Period between consecutive P-VOPs			
N	Period between consecutive I-VOPs			
Range	Search range for motion vectors			

Table 1: Parameters of Simulations

The measure of quality here is the PSNR, which is defined to be the Peak Signal-to-Noise Ratio of the luminance (Y) component of the decoded video compared to the input video at the same resolution. While PSNR is the quality measure used here, other possible measures include, e.g., MSE (Mean Square Error) and a Perceptual Distortion Measure. PSNR as used here is defined to be 20*log10(MSE) (dB).

Single Layer Coding

Single layer coding results for the QCIF and CIF Carphone sequences are shown in FIG. 5 and FIG. 6. The legend "150-30fps" indicates a sequence of 150 frames at 30 frames per second. Note that 150 frames was an arbitrarily chosen length and many suitable sequence lengths could be used. Conceptually, switching between spatial scalability and simulcast coding can occur as frequently as at every picture. However, this may result in syntax problems. Switching between groups of pictures (GOPs) is a realistic possibility.

Moreover, here and in the other figures, the bit rate is an average bit rate over a sequence. The circles denote the empirical results and the dotted lines 500, 600, respectively, represent logarithmic fits to the data using the following model:

$$\mathrm{PSNR} = A\ln(\mathrm{Bitrate}) + B .$$

The model allows each single layer to be represented by two parameters (constants) A and B (along with the range of bit rates where this model is valid). "ln" denotes the natural logarithm. "Bitrate" is the "x" parameter in the figures.

Simulcast Coding

A typical scenario encountered when transmitting multiple sequences is a constraint on the total bandwidth.

FIG. 7 is a plot of PSNR vs. fraction of total bits allocated to the lower-resolution stream, for a

total bandwidth of 0.29, 0.32, and 0.35 Mbps for QCIF/CIF simulcast. The figure shows examples of the different qualities that can be achieved by limiting the total bandwidth to 0.29, 0.32 and 0.35 Mbps, respectively, for simulcast ("Sim.") transmission of the Carphone QCIF and CIF sequences. Both qualities are plotted as functions of the fraction of total bits allocated to the lower-resolution stream (i.e., the QCIF stream). In particular, the solid lines 700, 710, 720 represent the PSNR of the lower-resolution (QCIF) sequence for total bit rates of 0.29, 0.32, and 0.35 Mbps, respectively. The dotted lines 750, 760, 770 represent the PSNR of the higher-resolution (CIF) sequence for total bit rates of 0.29, 0.32, and 0.35 Mbps, respectively.

Note the monotonicity of the data for both layers. That is, the PSNR either increases or decreases steadily (without a peak or valley).

While some of the figures refer to the fraction of bits that are allocation to the base layer or enhancement layer (for SS), or to the lower-resolution stream or higher-resolution stream (for Sim.), note that these values can also be expressed in terms of an absolute number of bits, a percentage, a fraction or percentage of a fixed reference value, or any other linear or non-linear metric or scale.

The independence of the single streams in simulcast coding causes an improvement in quality with

an allocation of more bits to the QCIF stream, and degradation of the other, (CIF) stream. This makes sense since the additional bits that are given to one stream are taken away from the other stream to maintain the same total bandwidth.

The curve crosses in the figures denote the point where the functions intersect, i.e., the point where the PSNR of both streams is equal. This point can be found using the logarithmic fits to the data of each stream and a bisection algorithm since the PSNR of each stream is a monotonic function of the bit rate of the lower-resolution stream.

In particular, the monotonicity of the PSNR for both layers implies that if the functions intersect, they will intersect at only one point. In some cases, there may be no intersection point due to insufficient or excess total bandwidth. An additional exit condition checking the difference in bit rates can be used to determine if there is no intersection. This has been omitted from the following algorithm for simplicity. Therefore, the algorithm below assumes that a proper total bit rate has been selected allowing the functions to intersect. A bisection algorithm using the difference in PSNR between the layers can be used to find the point of equal quality. An example algorithm follows:

1. Assume fixed total bitrate R_{TOT} and a threshold for convergence $T > 0$.

2. Let R_1 and R_2 be the minimum and maximum bitrates for single layer coding of the lower layer.

3. Let $R_3 = \frac{R_1 + R_2}{2}$.

4. Compute $PSNR_{lower}$ and $PSNR_{upper}$ for R_3 using the logarithmic fits to each layer:

$$PSNR_{lower} = A_{lower} \ln(R_3) + B_{lower}$$
$$PSNR_{upper} = A_{upper} \ln(R_{TOT} - R_3) + B_{upper}$$

where A_{lower} and B_{lower} are the parameters for the lower (lower-resolution) layer and A_{upper} and B_{upper} are the parameters for the upper (higher-resolution) layer.

5. Let DIFF = $PSNR_{lower}$ - $PSNR_{upper}$.

6. If the absolute value of DIFF is less than T,

the algorithm is finished and R_3 is the lower layer bitrate to achieve equal quality in both layers of simulcast coding.

Otherwise,

Set $R_1 = R_3$ if DIFF < 0.

Set $R_2 = R_3$ if DIFF > 0.

Go back to step 3.

This algorithm is just one example of how the crossover point can be found. Other techniques are possible, such as a linear interpolation.

FIG. 8 gives an example of the iterations needed to obtain the point 800 of equal quality in both streams of the QCIF/CIF simulcast coding of the

carphone sequence, for the 0.32 Mbps case. $R_1(i)$ and $R_2(i)$ represent the bit rates R_1 and R_2, respectively, for the lower-resolution stream at iteration i of the algorithm. Note that FIG. 8 shows the absolute bit rate of the lower-resolution stream, whereas FIG. 7 shows the fraction of the total bits that are allocated to the lower-resolution stream. It should be easily understood here and elsewhere how to convert between these two types of scales.

The point where the PSNRs intersect in FIG. 7 can be interpreted as the bit allocation where both streams are coded at approximately the same quality since PSNR is normalized with respect to picture size. Note that this point may not occur with some total bandwidths due to the limited dynamic range of each stream's coding.

The ability to transmit two streams of different resolutions that have roughly the same quality may be desirable in applications such as streaming video over the Internet with the resolutions used here. However, other applications may have different requirements. For example, consider two different agendas for the same simulcast system. One application may desire relatively higher quality in the lower-resolution stream to satisfy a larger number of receivers receiving the lower resolution. Another application may desire a relatively higher quality in the higher-resolution streams to satisfy the receivers receiving the higher resolution because of the higher cost of the

bandwidth required to receive the entire simulcast stream. The analysis in the following sections focuses on achieving the same quality for both streams, but this may not be the goal of some applications and this issue is revisited hereinbelow.

Additionally, FIG. 7 illustrates that the fraction of bits allocated to the lower-resolution stream to achieve equal quality in the two streams is essentially independent of the total bit rate. This is a very useful result and its significance can be seen by the following example. Assume that a multiplexer is combining simulcast bitstreams and has already determined the proper bit allocation between streams. The preceding result suggests that the multiplexer does not have to redetermine the proper bit allocation when reacting to a change in the total bandwidth. Instead, the proportion of bits allocated to each stream should remain the same.

FIGs 9 and 10 show the results of performing the same analysis as described above for different sequences at a wide variety of constrained total bandwidths. In particular, FIG. 9 shows the PSNR versus bitrate for the test sequences Basket 900, Bus 910, Carphone 920, Foreman 930, News 940, and Silentvoice 950.

The quality of both the lower-resolution (QCIF) and higher-resolution (CIF) streams is shown by only one curve for each sequence since the data in this plot

was obtained by requiring equal PSNR for both streams, and the bitrate shown is the total bitrate. For example, for the Basket sequence 900, when the total bitrate is 1 Mbps, both streams have a PSNR of approximately 24 dB. As the total bitrate is slowly increased, the PSNR of both streams increases, up to approximately 32 dB when the total bit rate is 4.5 Mbps.

FIG. 10 shows the fraction of total bits allocated to the lower-resolution streams versus total bitrate for the test sequences Basket 1000, Bus 1010, Carphone 1020, Foreman 1030, News 1040, and Silentvoice 1050. The data show that approximately 40 ± 4 % of the total bandwidth should be allocated to the lower-resolution stream to achieve equal quality in both streams of simulcast coding. This result can be used as a general guide for the bit allocation of simulcast bitstreams.

Spatial Scalable Coding

Results for the spatial scalable ("SS") coding of the Carphone QCIF and CIF sequences are shown in FIG. 11. The abscissa of each data point (on the horizontal axis) is the total bit rate of both streams, and the ordinate (on the vertical axis) represents the PSNR of the enhancement layer (the higher, CIF resolution) using spatial scalability. Each set of curves uses the same lower-resolution QCIF base layer (and therefore base layer bit rate and PSNR) with different enhancement layer bit rates. The circles denote

empirical results, and the dotted lines are logarithmic fits to each set of curves using the same base layer. Table 2 lists the bit rate (in Mbps) and PSNR of the QCIF base layer as well as the parameters (A and B) of the logarithmic fit ($Y = A \ln(X) + B$) for each set of curves.

Base Bitrate (Mbps)	Base PSNR	A	B
0.286	36.46	9.40	41.34
0.171	34.11	6.89	40.90
0.108	32.73	5.70	40.45
0.0835	31.69	5.36	40.26
0.0714	31.04	5.32	40.26
0.0625	30.46	5.36	40.34
0.0561	29.92	5.40	40.41
0.0518	29.48	5.47	40.49
0.0492	29.09	5.55	40.53
0.0472	28.76	5.62	40.61
0.0456	28.44	5.66	40.62

Table 2: Base (QCIF) Layer Characteristics of Spatial Scalable Coding for QCIF/CIF (Carphone)

In FIG. 11, the 0.286, 0.171, and 0.108 base layer bit rates are shown at curves 1100, 1110, and 1120, respectively. The remaining bit rates of 0.0835 through 0.0456 are shown at the aggregate curves 1130.

A constraint on the total bandwidth is examined for spatial scalable coding of the Carphone QCIF and CIF sequences. FIG. 12 is a plot of PSNR vs. the fraction of total bits allocated to the base layer, for

total bandwidths of 0.29, 0.32 and 0.35 Mbps for QCIF/CIF spatial scalability. The figure shows examples of the different qualities that can be achieved by limiting the total bandwidth to 0.29, 0.32 and 0.35 Mbps. The solid lines represent the PSNR of the base layer (QCIF) sequence and the dotted lines represent the PSNR of the enhancement layer (CIF) sequence. In particular, the solid lines 1200, 1210, 1220 represent the PSNR of the QCIF sequence for 0.29, 0.32, and 0.35 Mbps, respectively. The dotted lines 1250, 1260, 1270 represent the PSNR of the CIF sequence for 0.29, 0.32, and 0.35 Mbps, respectively.

Note the PSNR of the CIF sequences is not a monotonic function of the fraction of bits allocated to the base layer. That is, there is a peak in the PSNR for the CIF sequences near 20%.

As expected, the PSNR of the QCIF sequence is a monotonically increasing function of the fraction of total bits allocated to it. Moreover, it appears that allocating less than approximately 20% of the total bitstream to the base layer gives declining performance in both layers. The decline is relatively slight for the CIF layer, but rather sharp for the QCIF layer. This trend is also present with the other test sequences. This result is different from the one seen in the simulcast approach, where both layers are independent, and is due to the dependence of the CIF enhancement layer on the upsampled QCIF base layer.

This result of 20% allocation for the base layer can be a useful minimum boundary for the allocation of base layer bits for spatial scalable encoders.

Additionally, FIG. 12 shows that the fraction of bits allocated to the base layer to achieve equal quality in the two layers is essentially independent from the total bit rate. This result is similar to the conclusions obtained after analysis of the simulcast experiments and can be very useful for allocating spatial scalable bitstreams.

FIGs 13 and 14 show the results of performing the same analysis as described above for different sequences at a wide variety of constrained total bandwidths, where the base layer and enhancement layer have the same PSNR for a given total bit rate. In particular, FIG. 13 shows the PSNR versus total bitrate for the test sequences Basket 1300, Bus 1310, Carphone 1320, Foreman 1330, News 1340, and Silentvoice 1350. FIG. 14 shows the fraction of total bits allocated to the base layer versus total bitrate for the test sequences Basket 1400, Bus 1410, Carphone 1420, Foreman 1430, News 1440, and Silentvoice 1450. The data of FIG. 14 shows that approximately 45 ± 5 % of the total bandwidth should be allocated to the base layer to achieve equal quality in both layers of spatial scalable coded bitstreams. This percentage can be used as a general guide for the bit allocation of spatial scalable bitstreams.

FIGs 15 through 20 show the simulcast and spatial scalability results for each test sequence, where the QCIF and CIF video sequences have the same PSNR for a given total bit rate.

In particular, FIG. 15 shows the PSNR for spatial scalability 1500 and simulcast 1510 for the carphone sequence, FIG. 16 shows the PSNR for spatial scalability 1600 and simulcast 1610 for the news sequence, FIG. 17 shows the PSNR for spatial scalability 1700 and simulcast 1710 for the foreman sequence, FIG. 18 shows the PSNR for spatial scalability 1800 and simulcast 1810 for the basket sequence, FIG. 19 shows the PSNR for spatial scalability 1900 and simulcast 1910 for the silentvoice sequence, and FIG. 20 shows the PSNR for spatial scalability 2000 and simulcast 2010 for the bus sequence.

The operating regions of simulcast coding and spatial scalability are often different, with spatial scalability being the only option at relatively low total bit rates, and simulcast coding at relatively high total bit rates. Specifically, at relatively low bit rates, spatial scalability can be used if a coarse quantizer is used for residual coding. Simulcast coding may not be possible since the bandwidth may be too low to encode sequences at the higher resolution even with the coarsest quantizer. Note that much more information must be encoded for the simulcast case

since no information is available from the base layer. That is, there usually is a lot less signal energy in the residual (the difference between the uncoded enhancement layer and an upsampled version of the decoded base layer) than in the original signal. The enhancement layer of spatial scalability can be thought of as encoding the residual while the second, higher-resolution stream of simulcast coding is encoding the original, high resolution signal. Since we are using the same range of quantizers for both the enhancement layer of spatial scalability and the second layer of simulcast coding, it is not surprising that there are different ranges for the coded bitrates for the two methods.

Except for some regions with the Basket sequence, there is an improvement in quality gained by using spatial scalability at bit rates where both simulcast coding and spatial scalability are possible. Table 3 lists the range of PSNR improvements for each sequence using spatial scalability where both simulcast and scalable coding are possible. The negative value for the minimum PSNR improvement for the Basket sequence indicates that simulcast coding achieves higher quality video for part of the common operating region.

Note that the decision between simulcast coding and spatial scalability for a commercial application generally involves more than looking at the differences in PSNR or other quality measure. The lower layer bit

rate is smaller for simulcast coding, which favors simulcast coding since less bandwidth is required for reception of only the base layer. Additional decoder complexity is also required to decode spatial scalability bitstreams.

Sequence	Maximum PSNR Improvement Using Spatial Scalability	Minimum PSNR Improvement Using Spatial Scalability
Carphone	0.74	0.41
News	1.06	0.79
Foreman	0.75	0.68
Basket	0.71	-0.25
Silentvoice	1.27	1.18
Bus	0.65	0.11

Table 3: Range of PSNR Improvements Using Spatial Scalability Where Both Simulcast and Scalable Coding Are Possible

The previous analysis focused on achieving equal PSNR in both layers. As discussed earlier, this may not be the goal of some applications. A different view of this operating point reveals some additional insight into the general differences between simulcast coding and spatial scalability.

FIGs 21 through 23 show the results of both simulcast and scalable coding for the QCIF and CIF Carphone sequences with fixed total bandwidths of 0.29, 0.32 and 0.35 Mbps, respectively. The results are plotted as functions of the fraction of bits allocated

to the base layer (for SS), or the lower-resolution stream (for Sim.).

In particular, FIG. 21 shows the PSNR for 0.29 Mbps for simulcast CIF 2100, spatial scalability CIF 2110, and QCIF 2120 (which is the same for simulcast or spatial scalability). FIG. 22 shows the PSNR for 0.32 Mbps for simulcast CIF 2200, spatial scalability CIF 2210, and QCIF 2220. FIG. 23 shows the PSNR for 0.35 Mbps for simulcast CIF 2300, spatial scalability CIF 2310, and QCIF 2320.

Note that simulcast ("Sim.") outperforms spatial scalability ("SS") if a relatively small percentage of the total bit rate is assigned to the base layer. One general trend appears to be the increasing advantage of spatial scalability with more bits allocated to the base layer.

As an example, FIG. 21 provide visual markers to aid the following discussion. Point A' is the operating point for equal PSNR in both streams using simulcast coding. Point B' is the corresponding point for spatial scalability using the same amount of bits allocated to the base layer as Point A'. Point C' is the operating point for equal PSNR in both layers using spatial scalability. Note that the use of the operating points that achieves equal quality in both streams or layers (Points A' and C', respectively) causes different amounts of bits to be allocated to the lower-resolution stream or base layer depending,

respectively, on whether simulcast (Point A') or spatial scalability (Point C') is used.

Comparison of simulcast coding and spatial scalability with the same amount of bits allocated to the lower-resolution stream and base layer (Points A' and B', respectively) shows that spatial scalability results in a higher PSNR at the higher CIF resolution. Note that the PSNR at the lower QCIF resolution is the same for both simulcast coding and spatial scalability. Moreover, this is different from the gain in both the QCIF and CIF resolutions obtained by using the operating point for equal quality in both layers (Point C'). One way to interpret this concept is that part of the PSNR gain in the enhancement layer by using spatial scalable coding can be "exchanged" for an increase in the base layer by "moving" bits from the enhancement to the base layer. In fact, more bits can also be "moved" from the base layer to the enhancement layer. This concept can be visualized by simultaneously moving along the QCIF and CIF curves in FIG. 21. This allows different distributions, such as the points between Points B' and C' (for SS), or between Points A' and C' (for Sim.), to be achievable.

In general, the bit allocation problem involves two bit rates (the base and enhancement bitstreams for SS., and the lower-resolution and higher-resolution bitsreams for Sim.), and the choice between simulcast coding and spatial scalability. Note that more than

two streams or layers may be used, in which case the bit allocation problem is extended accordingly.

One method to obtain this three-dimensional data (assuming two streams or layers) is to fix the rate of the base layer (or lower-resolution stream), and decide between simulcast coding and spatial scalability for different bit rates for the enhancement layer or higher-resolution stream. By combining the data at different base layer (lower-resolution stream) rates, the complete three-dimensional data can be constructed.

FIGs 24 and 25 are examples of fixing the bit rate (and therefore, PSNR) of the base layer (or lower-resolution stream), for the transmission of the Carphone QCIF and CIF sequences. In particular, FIG. 24 shows the PSNR of the higher-resolution stream or enhancement layer, respectively, for simulcast 2400 and spatial scalability 2410 at 0.29 Mbps, and FIG. 25 shows the corresponding PSNR for simulcast 2500 and spatial scalability 2510 at 0.05 Mbps. Note that the curve for spatial scalability has a smaller dynamic range.

In FIG. 24, fixing the QCIF data results in a PSNR of 36.45 for that data. The total bit rate is then 0.29 Mbps + the enhancement layer or higher-resolution stream layer bit rate. The data point at (0 Mbps, 30 dB) results if no enhancement layer data is used, i.e., the base layer is simply upsampled (using bilinear interpolation) to create the enhancement layer.

In FIG. 25, the fixed QCIF bit rate results in a PSNR of 28.36 for that data. The total bit rate is then 0.05 Mbps + the enhancement layer or higher-resolution stream bit rate. The data point at (0 Mbps, 27 dB) results if no enhancement layer data is used, i.e., the base layer is simply upsampled (using bilinear interpolation) to create the enhancement layer. The general trend is for spatial scalability to be more efficient at lower enhancement layer/higher-resolution stream bit rates, while simulcast coding is more efficient at higher enhancement layer/higher-resolution stream bit rates. An important result is to determine the boundary where simulcast coding and spatial scalability are equivalent. This boundary can then be used to determine whether one should use simulcast coding or spatial scalability.

The functions for simulcast coding and spatial scalability may not intersect, but the curves can be extrapolated to find an intersection point by fitting both curves to logarithms and finding the intersection of the logarithmic fits. In particular, assume the two logarithmic functions are:

$$Y_1 = A \ln\left(X\right) + B$$

$$Y_2 = C \ln\left(X\right) + D$$

Y1 is the PSNR for one curve, e.g., the simulcast coding curve, with curve fit constants A and B. Y2 is the PSNR for the other curve, e.g., the spatial

scalability curve, with curve fit constants C and D. Solving for the point of intersection between the two logarithmic functions yields:

$$Y_1 = Y_2$$
$$A\ln(X) + B = C\ln(X) + D$$
$$(A - C)\ln(X) = D - B$$
$$\ln(X) = \frac{D - B}{A - C}$$
$$X = e^{\frac{D-B}{A-C}}$$

The crosses in the figures (point 2420 in FIG. 24, and point 2520 in FIG. 26) represent the estimated points of intersection.

FIG. 26 shows the results of applying this technique of estimating the boundary between simulcast coding and spatial scalability for all the test sequences. This figure shows the test sequences Basket 2600, Bus 2610, Carphone 2620, Foreman 2630, News 2640, and Silentvoice 2650. For each video sequence, the data indicates that operating points above the curve should use simulcast coding and points below the curve should use spatial scalability to obtain the highest PSNR for the CIF video sequence (e.g., the enhancement layer or higher-resolution stream). Note that specifying the base bit rate uniquely determines the PSNR of the QCIF sequence since this resolution is single layer coded. The decision boundaries tend to have the same shape with different scales.

FIG. 27 demonstrates the similarity of the decision boundaries this by normalizing the base layer and the enhancement layer bit rates (for SS), and by normalizing the lower-resolution stream and higher-resolution stream bit rates (for Sim.), using the range of the data and the following formula:

$$Normalized\ Bitrate = \frac{Bitrate - Minimum}{Maximum - Minimum}.$$

This formula maps the minimum absolute bit rate to a zero normalized bit rate and the maximum absolute bit rate to a normalized bit rate of one.

FIG. 27 shows normalized decision boundaries for the test sequences Basket 2700, Bus 2710, Carphone 2720, Foreman 2730, News 2740, and Silentvoice 2750.

The results above allow one to choose between simulcast and spatial scalability strictly on the basis of which mode provides higher quality. Considering other issues such as the additional receiver complexity required for spatial scalability may require not only determining which mode is better, but how much improvement is obtained, especially when factors support the other mode. This requires looking at the three-dimensional data. Note that determining which mode is better regardless of the PSNR difference is equivalent to a projection of the three-dimensional data onto a two-dimensional space. Visualizing and establishing decision criteria for surfaces is

difficult for general scenarios and may be best handled in a case-by-case manner.

Multicast environment

The present invention can be applied to video broadcasting in a multicast environment, such as a video server providing video (e.g., movies) to users via a computer network such as the Internet. The users may receive the data via an appliance such as a personal computer, Internet-capable set-top box, or the like. In this environment, multiple clients require different types of service due to variations in their processing power, memory resources and available bandwidth. The server would like to provide different resolutions and/or qualities of the same video sequence to satisfy each type of client. Note that the server should provide content for all the service rates continuously, otherwise, an entire client type does not receive service.

This scenario is different from the unicast environment, where different levels of service are provided to account for dynamic changes in the point-to-point transmission. In this case, the server can adaptively switch between bitstreams to provide service commensurate with the available resources. Therefore, only one bitstream is transmitted at any instance and it can be tailored to the target bit rate. This scheme provides high quality video, and benefits from low

decoder complexity since single-layer decoding is always used.

Different services may involve improving the spatial resolution, temporal resolution and/or quality of the video transmission with increasing bit rates. This discussion focuses on providing different levels of spatial resolution. Consider the following example with two levels of service. Clients at the lower service rate receive QCIF resolution video and those at the higher service rate receive CIF resolution video. Both service rates receive video with the same temporal resolution.

One approach to providing multicast service is to simulcast code the sequence at all of the service rates. This approach produces high quality video at all service rates. However, since service must be provided to all client types, the encoder must transmit a large amount of data, i.e., the sum of all the service rates. The main expense in multicast transmission is the total used bandwidth (in terms of total number of packets transmitted). Network congestion is also a problem, so it may be necessary to constrain the total combined rate of all the service rates. The minimum bandwidth necessary is the largest single service rate that is able to provide adequate service to all clients. The following discussion assumes that a constraint on the total bandwidth is imposed where the constrained rate is between the

largest single service rate and the sum of all the service rates.

The simulcast approach can still be used to provide two levels of service by reducing the higher bitstream to the difference between the constrained total and the lower bitstream. Another approach is to use spatially scalability. Intuitively, one expects spatial scalability to perform better since the information in the base layer (QCIF) sequence is used to assist construction of the enhancement layer (CIF) sequence. In general, this is true, but it has been found in connection with the present invention that simulcast coding can outperform spatial scalability when a small proportion of bits are allocated to the base layer/lower-resolution stream. This is counterintuitive, since spatial scalability "reuses" information in the base layer and its enhancement layer has the same bit rate as the single layer bitstream used by simulcast coding. One explanation may be that the overhead incurred by using a scalable coding syntax surpasses the gain obtained. The base layer also may not provide good prediction when a small amount of bits is allocated to it.

FIGs 28 and 29 are examples of multicast broadcasting using simulcast coding and spatial scalability, respectively, when the total used bandwidth is constrained to 200 kbps and the user service rates are 50 kbps and 200 kbps.

In FIG. 28, a server 2800 provides simulcast-coded lower-resolution and higher-resolution streams at 40 and 160 kbps, respectively, to a 200 kbps channel 2810 and a switching device, such as a router 2820. The 40 kbps stream is routed to the first service via a path 2830, and the 160 kbps stream is routed to the second service via a path 2850. The router 2820 does not route the 160 kbps stream to the first service since the first service cannot handle this data rate. Moreover, the router 2820 does not route the 40 kbps stream to the second service since this service only has use for one of the streams, and can handle the higher resolution 160 kbps stream.

Any known computer-network routing protocol may be used to achieve this result. In particular, the router 2820 should be informed of which services can handle which data rates. Each service can represent many end users.

In FIG. 29, a server 2900 provides spatial scalability-coded base and enhancement layers at 40 and 160 kbps, respectively, to a 200 kbps channel 2910 and a switching device, such as a router 2920. The 40 kbps base layer is routed to the first service via a path 2930, and both the 40 kbps base layer and 160 kbps enhancement layer are routed to the second service via paths 2940 and 2950, respectively (which may be the same path). The second service receives both the

layers since they must be used together to obtain the maximum information (e.g., image resolution).

Note that the video for the 50 kbps service rate is identical regardless of which coding approach is used. That is, the user with the 50 kbps service receives only the 40 kbps lower-resolution stream when simulcast coding is used, or the equivalent 40 kbps base layer when scalability coding is used.

The 200 kbps service in the simulcast coding approach (FIG. 28) constructs video for the higher service rate using the 160 kbps stream. This service does not use the 40 kbps single layer stream since it is independent from the 160 kbps layer and therefore there is no benefit to using it.

However, the 200 kbps service in the spatial scalability approach (FIG. 29) can use the 40 kbps base layer in addition to the 160 kbps enhancement layer, allowing it to construct video for the higher service rate using 200 kbps. These figures support the intuition that spatial scalability can outperform simulcast coding.

As discussed, FIG. 22 shows the results of encoding the QCIF and CIF Carphone sequences using both spatial scalability and simulcast coding with different bit allocations, but a fixed total bandwidth of 0.32 Mbps. Note that spatial scalability outperforms simulcast coding for many different bit allocations. However, simulcast coding is more efficient if less

than about 20% of the total bandwidth is allocated to the base layer/lower-resolution stream.

FIG. 30 shows a decision boundary 2620 (from FIG. 26) between simulcast coding and spatial scalability for the Carphone QCIF and CIF sequences. In accordance with the invention, operating points above the decision boundary should use simulcast coding, and points below it should use spatial scalability coding. The definition of operating point used here is the average bitrate over the entire sequence. Note that similar analysis can be performed using a smaller subset of the sequence such as a group of pictures. The methods described in this document can be used to construct this curve and then used to determine whether simulcast coding or spatial scalability should be used to encode the video.

A fixed total bitrate gives only one constraint on two variables (the base and enhancement bit rates), therefore, one can use any distribution of the total bitrate. For example, assume that 0.5 Mbps total bitrate is available. In one case, 0.25 Mbps is used for the QCIF resolution, and 0.25 Mbps is used for the CIF resolution. Since this operating point 3000 is below the curve 2620, spatial scalability should be used to obtain the best quality for the CIF data. In another case, 0.1 Mbps is used for the QCIF resolution, and 0.4 Mbps is used for the CIF resolution. Since this operating point 3020 is above the curve 2620,

simulcast coding should be used to achieve the best quality for the CIF data. Recall that the quality of the QCIF data is the same regardless of whether simulcast coding or scalability coding is used.

While the simulations described here used a number of encoding and decoding experiments before settling on which transmission mode to use, one may be able to determine model parameters without having to run encoding/decoding experiments for every sequence.

Determining new model parameters for each sequence may not be necessary if one already has good model parameters. This may be the case for sequences that are similar. For example, in FIG. 26, the sequences carphone 2620, news 2640, and silentvoice 2650 have a similar coding decision boundary. Using the boundary obtained with one of these sequences to assist the simulcast/scalable coding decision may still be optimal for the coding of a different sequence, as long as the data points are not close to the decision boundary. Determining new model parameters for each sequence may be desirable, especially if one wants to maximize the quality of video delivery, despite the increased computational costs.

FIG. 31 illustrates an adaptive simulcast/spatial scalability encoder apparatus in accordance with the present invention. A higher-resolution video sequence (such as CIF), and a lower-resolution video sequence (such as QCIF) are provided to an analysis function

3110, and to respective switches 3130 and 3140. The analysis function 3110 analyzes the video sequences based on the above discussion to provide a select signal to the switches 3130, 3140 to route both of the sequences to either the simulcast encoder 400 or the scalable encoder 100. See Figures 1 and 4, respectively.

The analysis function 3110 may include a decoder and encoder, where the PSNR of a layer is determined by comparing the output of the decoder to the input to the encoder.

Successive sequences, each having several (e.g., 150) pictures/frames, may be analyzed to adaptively route each sequence to either the simulcast encoder 400 or the scalable encoder 100. Moreover, the analysis may occur off-line, prior to when the video is transmitted to a user. In this manner, unnecessary processing delays are avoided. For example, the video data may be recovered from a memory, analyzed, then returned to the storage device. Each sequence of analyzed data may be marked to indicate whether it is to be subsequently routed to either the simulcast encoder 400 or the scalable encoder 100. This marking may be achieved any number of ways, such as providing overhead control bits with the video data.

Moreover, note that the decoders should have the capability to determine whether they are receiving a simulcast-coded or scalability-coded stream. This can

be achieved according to the relevant data standard (e.g., MPEG-4). Moreover, the decoders should have scalable decoding capabilities and single layer decoding capabilities. Advantageously, the invention can be carried out without modification to such decoders.

It should now be appreciated that the present invention provides for the compression of video data for multicast environments. Spatial scalability and simulcast coding are used in the compression process. Simulations are provided to compare the performance of spatial scalability and simulcast coding of lower-resolution (e.g., QCIF) and higher-resolution (e.g., CIF) sequences. The main results and conclusions of this work are:

- The use of a logarithmic model to represent single layer coding results. This model can then be used to easily compute the bit allocation that achieves equal quality in both layers of simulcast coding.
- Allocating 40 ± 4 % (e.g., 36-44%) of the total bandwidth to the lower-resolution stream achieves equal quality in both streams of simulcast coding.
- Allocating less than 20% of the total bandwidth to the base layer for spatial scalable coding is inefficient.
- Allocating 45 ± 5 % (e.g., 40-50%) of the total bandwidth to the base layer achieves equal quality in both layers of spatial scalable coding.

- Spatial scalability may be the only option at relatively low enhancement bit rates, and simulcast coding may be the only option at relatively high enhancement bit rates, but spatial scalability usually gives higher quality in the common operating region.
- Decision boundaries can be generated to guide the decision between spatial scalability and simulcast coding.
- Simulcast coding can outperform spatial scalability when a small proportion of bits is allocated to the base layer/lower-resolution stream. This may be due to the overhead of the scalable coding syntax and/or bad prediction from the base layer.

Although the invention has been described in connection with various preferred embodiments, it should be appreciated that various modifications and adaptations may be made thereto without departing from the scope of the invention as set forth in the claims.

What is claimed is:

1. A method for coding video data, comprising the steps of:

modeling a quality of lower resolution video data, and of corresponding higher resolution video data, for varying bit allocations between the lower and higher resolution video data;

determining a crossover point of the modeled qualities wherein the lower and higher resolution video data have an approximately equal quality, and the corresponding bit allocation; and

coding the video data in accordance with the corresponding bit allocation.

2. The method of claim 1, wherein:

the video data coded in said coding step comprises a sequence of pictures.

3. The method of claim 1, wherein:

the corresponding bit allocation is an average over a sequence of pictures that comprise the lower and higher resolution video data.

4. The method of claim 1, wherein:

the video data coded in said coding step is different than the video data used in said modeling step.

5. The method of claim 1, wherein:

the video data coded in said coding step is the same as the video data used in said modeling step.

6. The method of claim 1, wherein:

the quality comprises a peak signal-to-noise ratio (PSNR).

7. The method of claim 1, wherein:

said modeling step uses a logarithmic model.

8. The method of claim 1, wherein:

the coded video data is coded in independent simulcast streams.

9. The method of claim 1, wherein:

the coded video data is coded using spatial scalability coding in a base layer and at least one enhancement layer.

10. The method of claim 1, wherein:

the crossover point is determined in accordance with a bisection algorithm.

11. The method of claim 1, wherein:

the crossover point is extrapolated.

12. A method for coding video data, comprising the steps of:

modeling a quality of lower resolution video data, and of corresponding higher resolution video data, for varying bit allocations between the lower and higher resolution video data, by using each of: (a) simulcast coding, and (b) spatial scalability coding; and

coding the video data at a desired bit allocation for the lower and higher resolution video data using one of the simulcast coding and the spatial scalability coding according to whichever results in a larger modeled quality at the desired bit allocation.

13. The method of claim 12, wherein:

the video data comprises a sequence of pictures.

14. The method of claim 12, wherein:

the desired bit allocation is an average over a sequence of pictures that comprise the lower and higher resolution video data.

15. The method of claim 12, wherein:

the video data coded in said coding step is different than the video data used in said modeling step.

16. The method of claim 12, wherein:

the video data coded in said coding step is the same as the video data used in said modeling step.

17. The method of claim 12, wherein:

the modeled qualities comprise a peak signal-to-noise ratio (PSNR).

18. The method of claim 12, wherein:

said modeling step uses a logarithmic model.

19. A method for coding video data, comprising the steps of:

(a) determining a bit allocation decision boundary for varying bit allocations for lower resolution video data;

wherein, for the varying bit allocations for the lower resolution data, the decision boundary designates a bit allocation for corresponding higher resolution video data;

(b) determining an operating point for the video data based on desired bit allocations for lower resolution video data thereof and corresponding higher resolution video data thereof;

(c) coding the video data using simulcast coding if the operating point indicates the desired bit allocation of the higher resolution video data of said

step (b) exceeds a bit allocation designated by the decision boundary; and

(d) coding the video data using simulcast coding if the operating point indicates the desired bit allocation of the higher resolution video data of said step (b) is less than a bit allocation designated by the decision boundary.

20. The method of claim 19, wherein:

the decision boundary corresponds to a bit allocation of approximately 20% of an available bandwidth to the lower resolution video data in said step (a) and approximately 80% of the available bandwidth to the higher resolution video data in said step (a).

21. The method of claim 19, wherein:

the video data comprises a sequence of pictures.

22. The method of claim 21, wherein:

the desired bit allocations in said step (b) are averages over the sequence of pictures.

23. The method of claim 19, wherein:

the video data coded in said coding steps is different from the video data used in said step (a).

24. The method of claim 19, wherein:

the video data coded in said coding steps is the same as the video data used in said step (a).

25. A method for coding video data comprising lower resolution data and corresponding higher resolution data, comprising the steps of:

(a) allocating a first portion of an available bandwidth to the lower resolution data, and a remainder of the available bandwidth to the higher resolution data; and

(b) coding the video data using simulcast coding if the first portion is less than approximately 20% of the available bandwidth, and using spatial scalability coding if the first portion is greater than approximately 20% of the available bandwidth.

26. A method for coding video data comprising lower resolution data and corresponding higher resolution data, comprising the step of:

allocating approximately 36-44% of an available bandwidth to the lower resolution data, and a remainder of the available bandwidth to the higher resolution data to achieve an approximately equal quality for the lower and higher resolution data when they are simulcast coded.

27. The method of claim 26, wherein:

the quality comprises a peak signal-to-noise ratio (PSNR).

28. A method for coding video data comprising lower resolution data and corresponding higher resolution data, comprising the step of:

allocating approximately 40-50% of an available bandwidth to the lower resolution data, and a remainder of the available bandwidth to the higher resolution data to achieve an approximately equal quality for the lower and higher resolution data when they are spatial scalability coded.

29. The method of claim 28, wherein:

the quality comprises a peak signal-to-noise ratio (PSNR).

30. A method for communicating video data comprising lower resolution data and corresponding higher resolution data to at least a first user via a network that adaptively uses spatial scalability coding and simulcast coding, comprising the step of:

configuring a switching device in the network to selectively forward at least one of the lower and higher resolution data to the first user according to which of the spatial scalability coding and the

simulcast coding is being used, and according to a data service capacity of the first user.

31. The method of claim 30, wherein:

when the simulcast coding is used, the switching device is configured to: (a) forward the higher resolution data to the first user if a data rate of the higher resolution data is within the data service capacity, and (b) to forward the lower resolution data to the first user if the data rate of the higher resolution data is not within the data service capacity, but a data rate of the lower resolution data is within the data service capacity.

32. The method of claim 30, wherein:

when spatial scalability coding is used, the switching device is configured to: (a) forward both the higher and lower resolution data to the first user if a combined data rate thereof is within the data service capacity, and (b) to forward the lower resolution data to the first user if the combined data rate is not within the data service capacity, but a data rate of the lower resolution data is within the data service capacity.

33. The method of claim 30, wherein:

the switching device comprises a router.

34. An apparatus for coding video data, comprising:

means for modeling a quality of lower resolution video data, and of corresponding higher resolution video data, for varying bit allocations between the lower and higher resolution video data;

means for determining a crossover point of the modeled qualities wherein the lower and higher resolution video data have an approximately equal quality, and the corresponding bit allocation; and

means for coding the video data in accordance with the corresponding bit allocation.

35. An apparatus for coding video data, comprising:

means for modeling a quality of lower resolution video data, and of corresponding higher resolution video data, for varying bit allocations between the lower and higher resolution video data, by using each of: (a) simulcast coding, and (b) spatial scalability coding; and

means for coding the video data at a desired bit allocation for the lower and higher resolution video data using one of the simulcast coding and the spatial scalability coding according to whichever results in a larger modeled quality at the desired bit allocation.

36. An apparatus for coding video data, comprising:

(a) means for determining a bit allocation decision boundary for varying bit allocations for lower resolution video data;

wherein, for the varying bit allocations for the lower resolution data, the decision boundary designates a bit allocation for corresponding higher resolution video data;

(b) means for determining an operating point for the video data based on desired bit allocations for lower resolution video data thereof and corresponding higher resolution video data thereof;

(c) means for coding the video data using simulcast coding if the operating point indicates the desired bit allocation of the higher resolution video data of said means (b) exceeds a bit allocation designated by the decision boundary; and

(d) means coding the video data using simulcast coding if the operating point indicates the desired bit allocation of the higher resolution video data of said means (b) is less than a bit allocation designated by the decision boundary.

37. An apparatus for coding video data comprising lower resolution data and corresponding higher resolution data, comprising:

(a) means for allocating a first portion of an available bandwidth to the lower resolution data, and a remainder of the available bandwidth to the higher resolution data; and

(b) means for coding the video data using simulcast coding if the first portion is less than approximately 20% of the available bandwidth, and using spatial scalability coding if the first portion is greater than approximately 20% of the available bandwidth.

38. An apparatus for coding video data comprising lower resolution data and corresponding higher resolution data, comprising:

means for allocating approximately 36-44% of an available bandwidth to the lower resolution data, and a remainder of the available bandwidth to the higher resolution data to achieve an approximately equal quality for the lower and higher resolution data when they are simulcast coded.

39. An apparatus for coding video data comprising lower resolution data and corresponding higher resolution data, comprising:

means for allocating approximately 40-50% of an available bandwidth to the lower resolution data, and a remainder of the available bandwidth to the higher resolution data to achieve an approximately equal

quality for the lower and higher resolution data when they are spatial scalability coded.

40. An apparatus for communicating video data comprising lower resolution data and corresponding higher resolution data to at least a first user via a network that adaptively uses spatial scalability coding and simulcast coding, comprising:

means for configuring a switching device in the network to selectively forward at least one of the lower and higher resolution data to the first user according to which of the spatial scalability coding and the simulcast coding is being used, and according to a data service capacity of the first user.

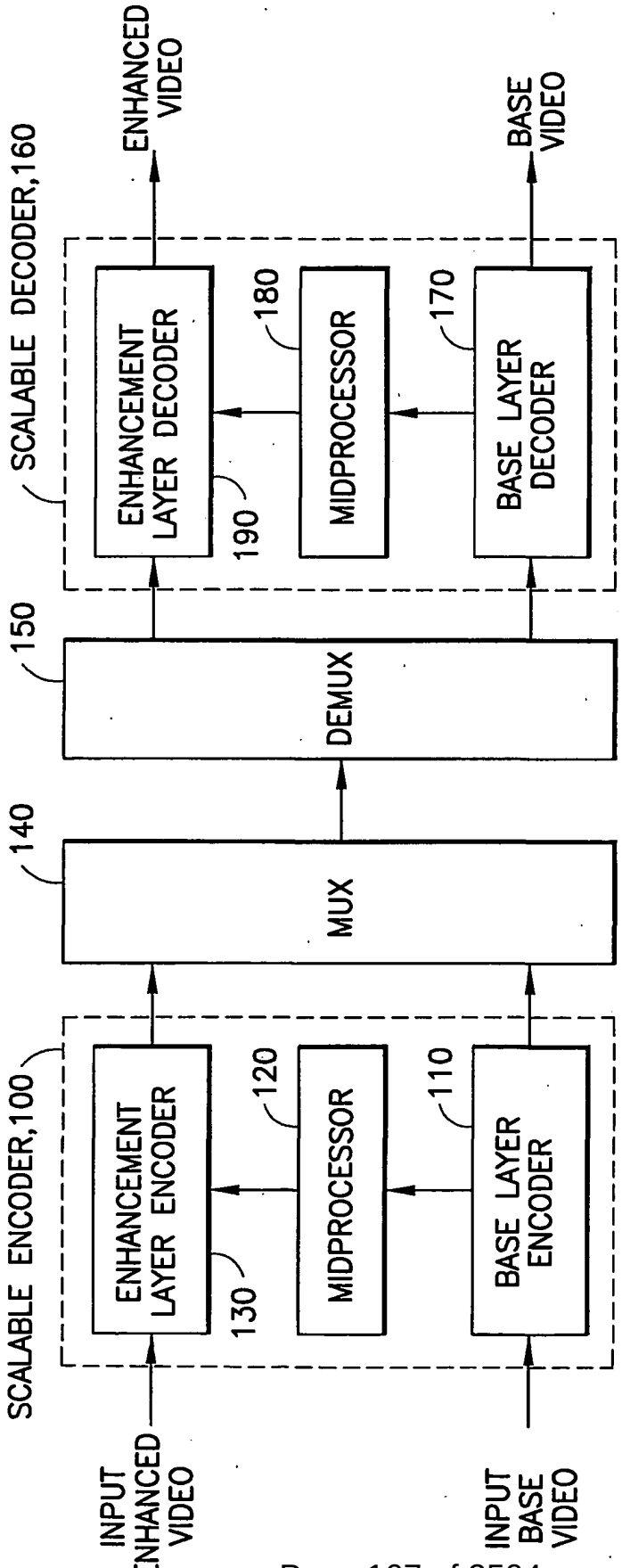

INPUT ENHANCED VIDEO
INPUT BASE VIDEO
SCALABLE ENCODER,100
ENHANCEMENT LAYER ENCODER
MIDPROCESSOR
BASE LAYER ENCODER
130
120
110
140
MUX
150
DEMUX
SCALABLE DECODER,160
ENHANCEMENT LAYER DECODER
MIDPROCESSOR
BASE LAYER DECODER
190
180
170
ENHANCED VIDEO
BASE VIDEO

FIG.1

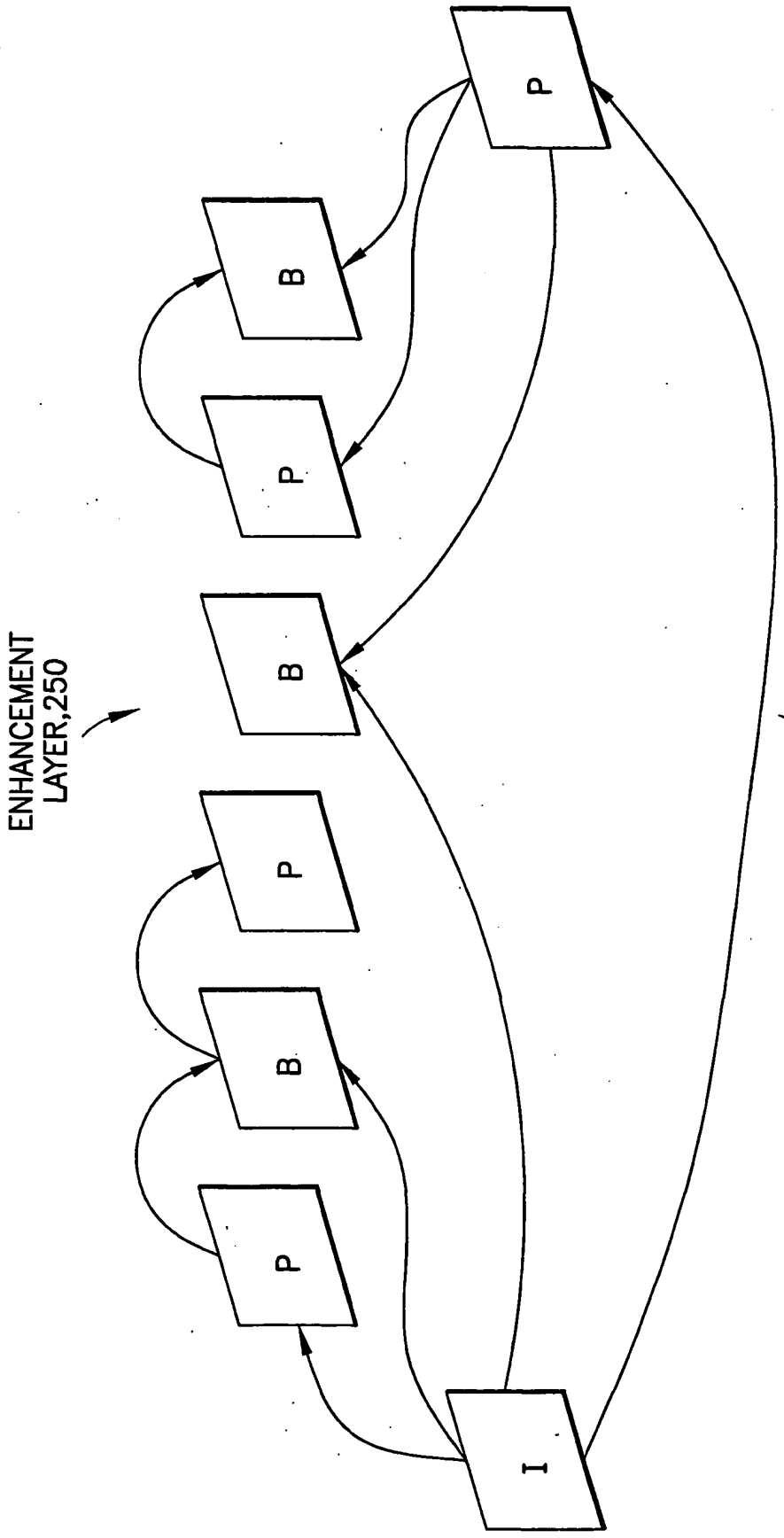


FIG.2

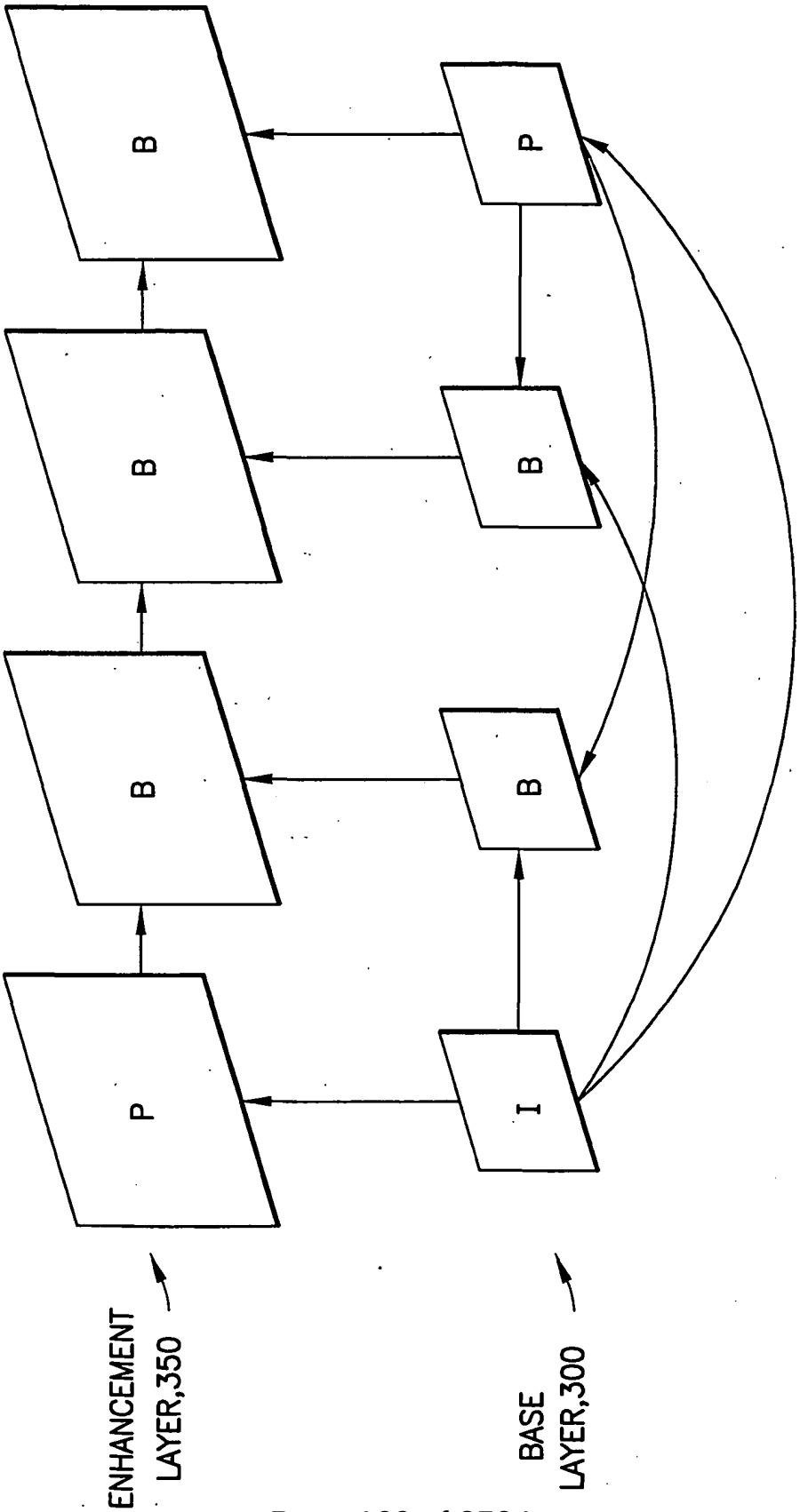


FIG.3

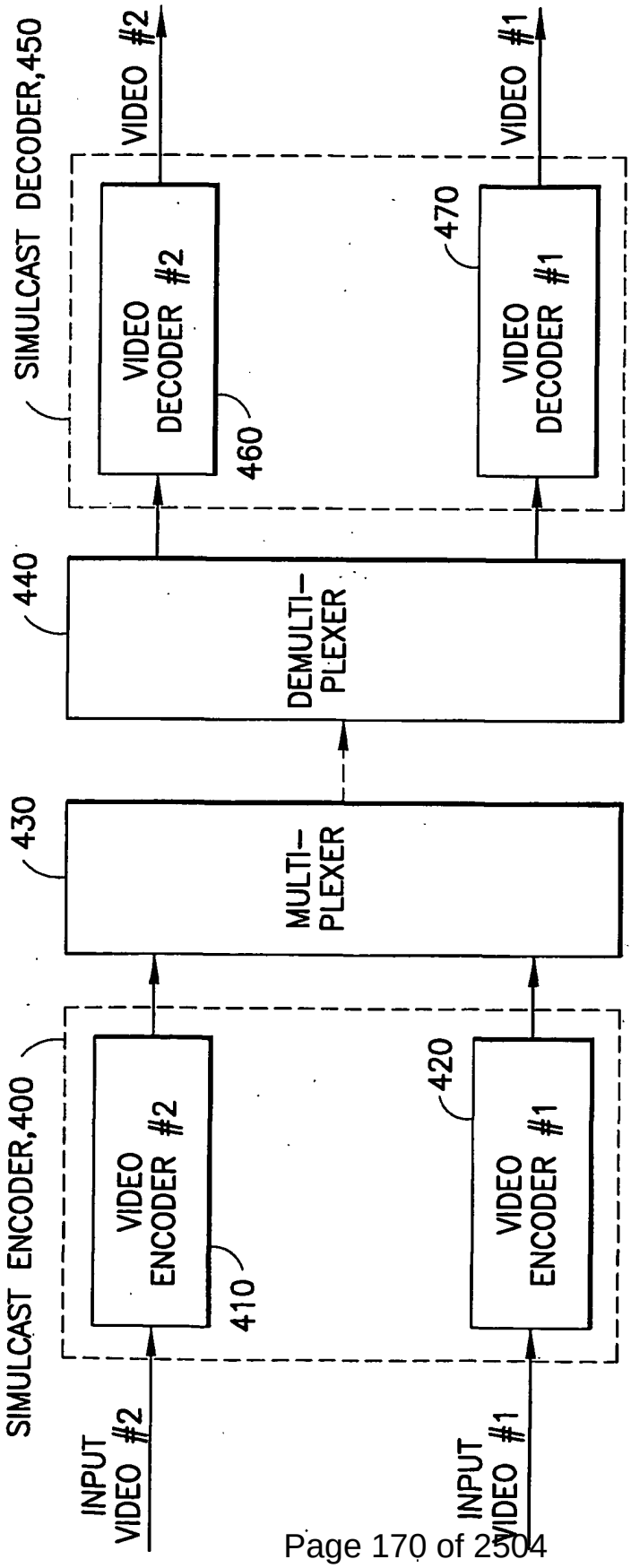

SIMULCAST ENCODER, 400
VIDEO ENCODER #2
410
VIDEO ENCODER #1
420
INPUT VIDEO #2
INPUT VIDEO #1
430
MULTI-PLEXER
440
DEMULTI-PLEXER
SIMULCAST DECODER, 450
VIDEO DECODER #2
460
VIDEO DECODER #1
470
VIDEO #2
VIDEO #1

FIG.4

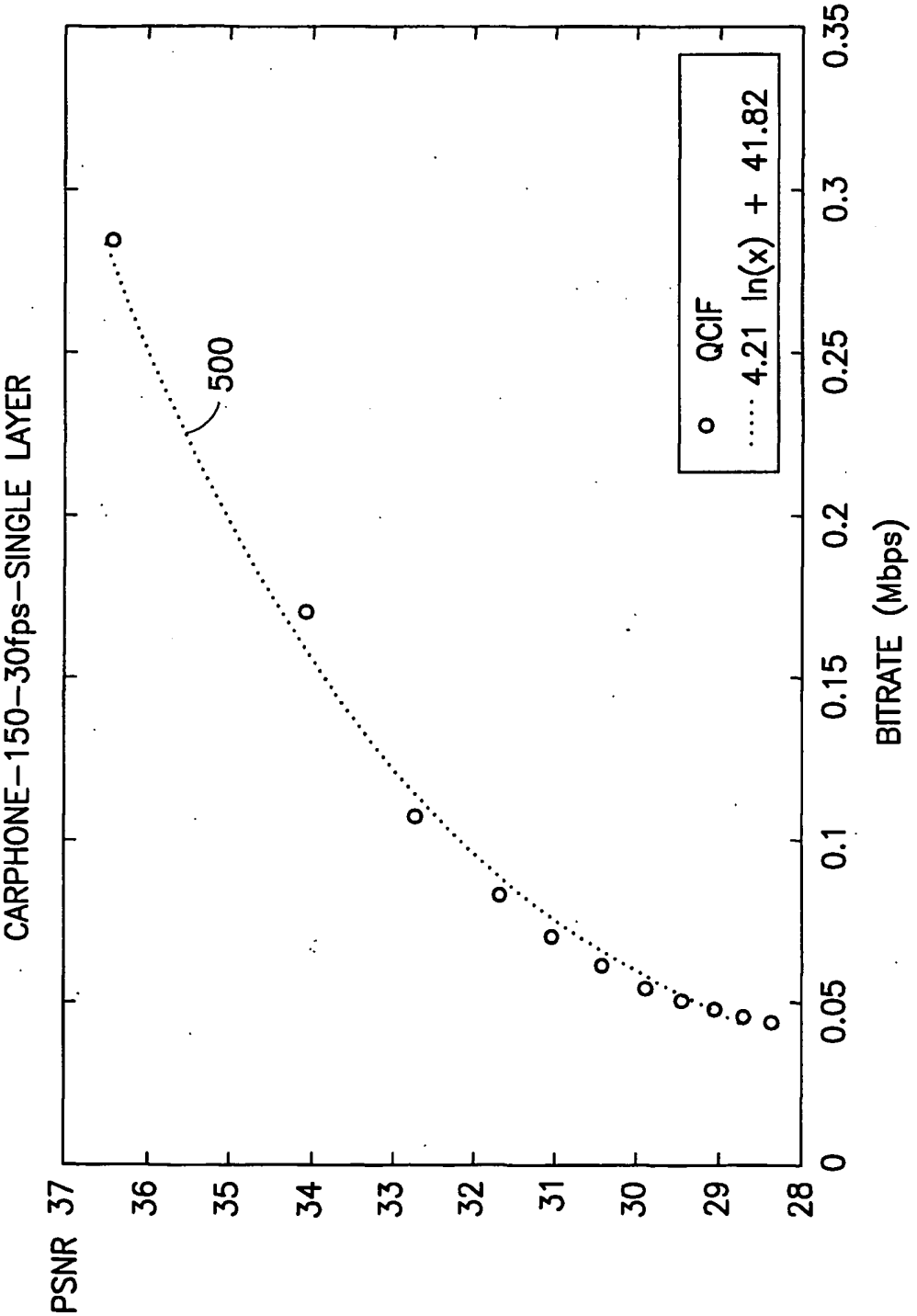

CARPHONE-150-30fps-SINGLE LAYER
PSNR
37
36
35
34
33
32
31
30
29
28
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
BITRATE (Mbps)
500
o QCIF
..... 4.21 ln(x) + 41.82

FIG.5

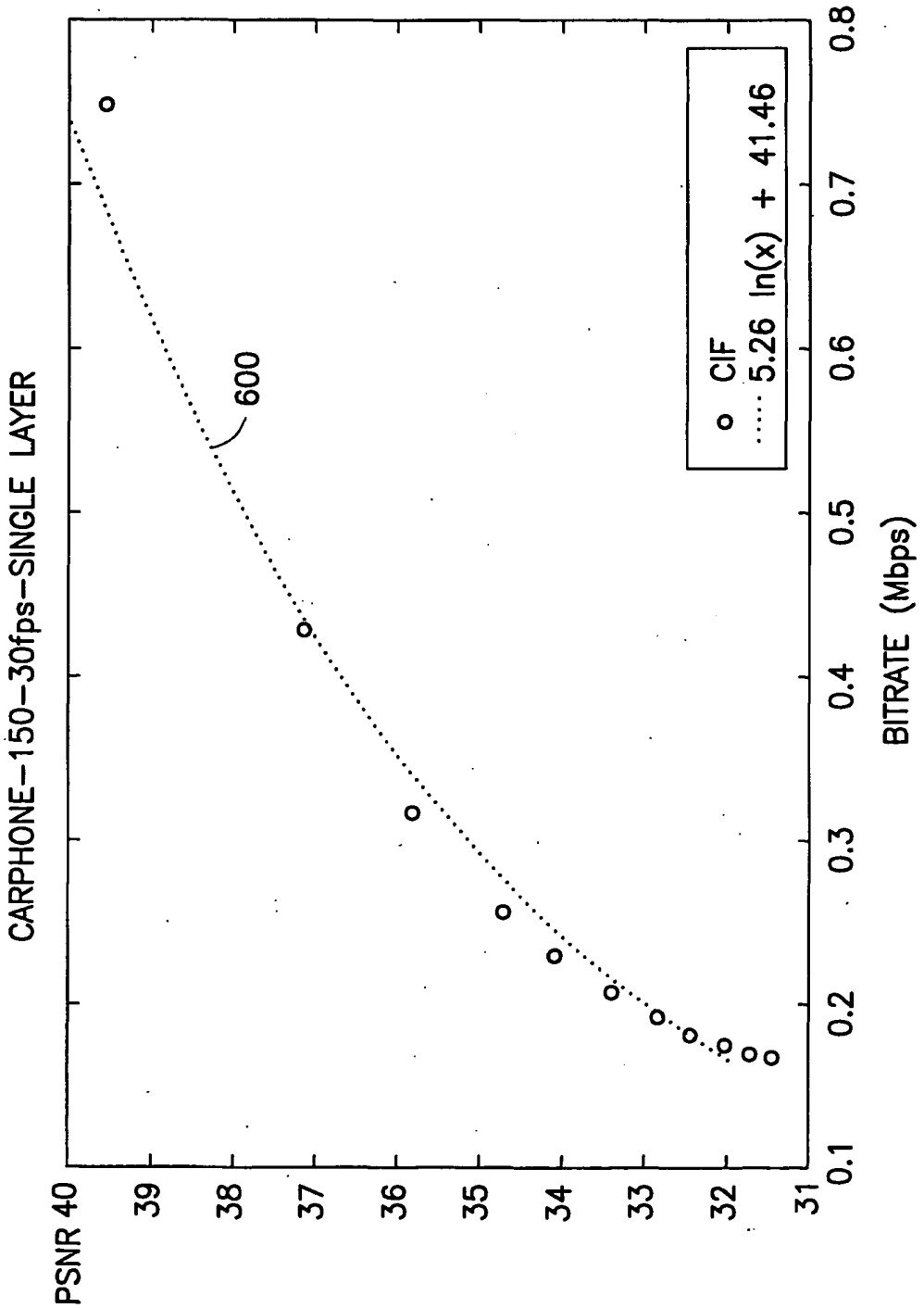

CARPHONE-150-30fps-SINGLE LAYER
PSNR 40
39
38
37
36
35
34
33
32
31
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
BITRATE (Mbps)
600
o CIF
..... 5.26 ln(x) + 41.46

FIG.6

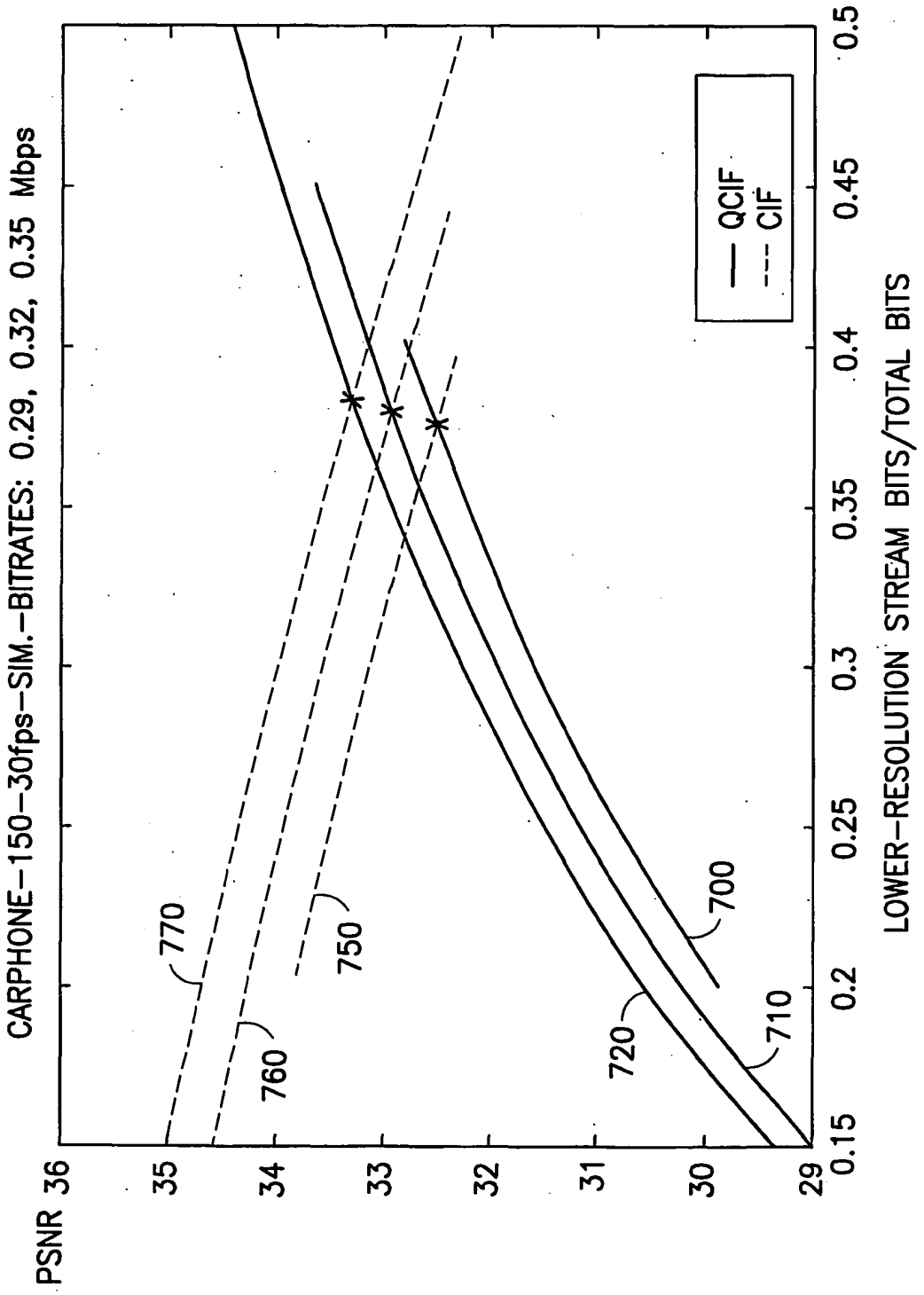

CARPHONE–150–30fps–SIM.–BITRATES: 0.29, 0.32, 0.35 Mbps
PSNR 36
35
34
33
32
31
30
29
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
LOWER–RESOLUTION STREAM BITS/TOTAL BITS
QCIF
CIF
770
760
750
700
720
710

FIG.7

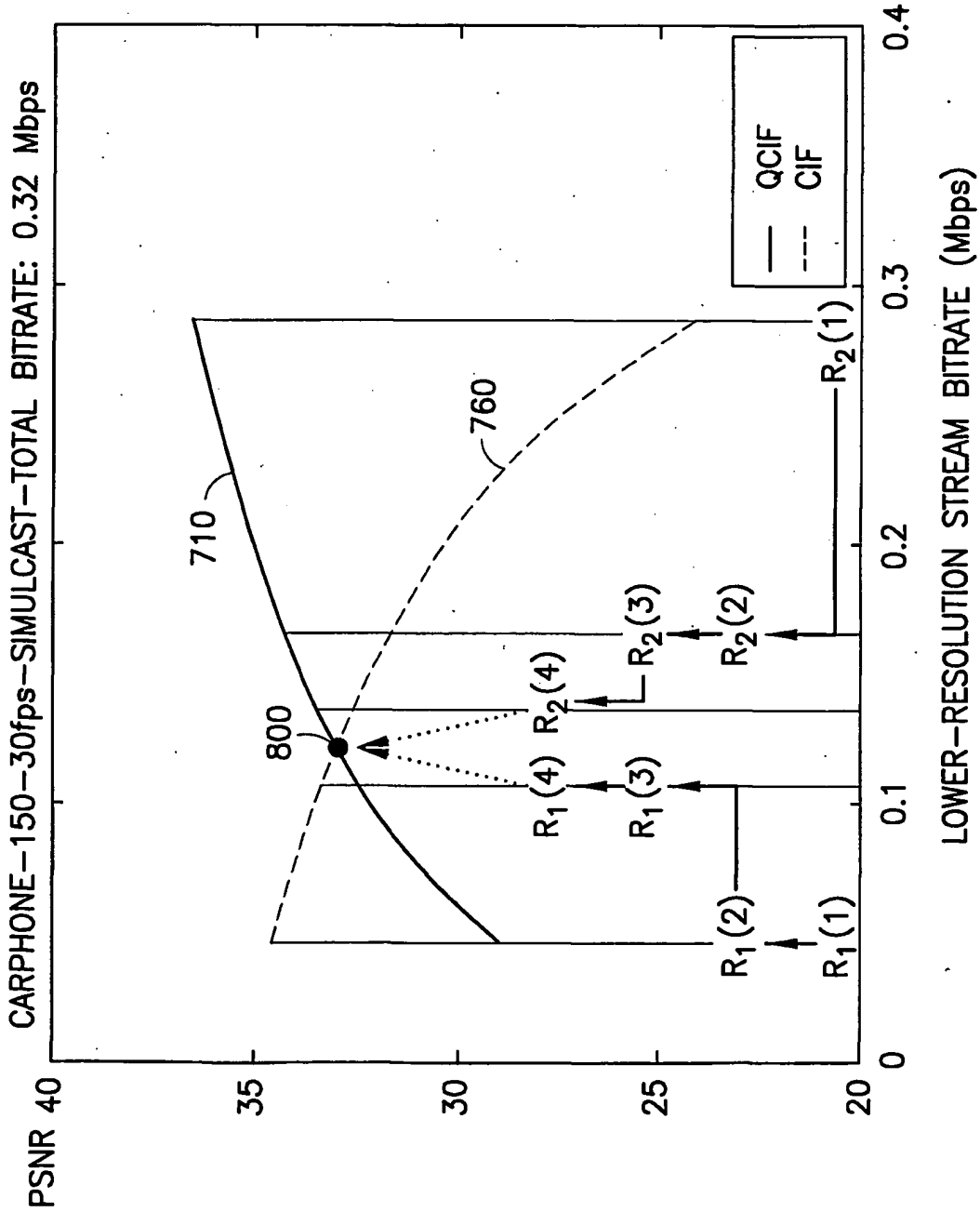

CARPHONE-150-30fps-SIMULCAST-TOTAL BITRATE: 0.32 Mbps
PSNR
40
35
30
25
20
0
0.1
0.2
0.3
0.4
LOWER-RESOLUTION STREAM BITRATE (Mbps)
QCIF
CIF
710
760
800
$R_2(1)$
$R_2(2)$
$R_2(3)$
$R_2(4)$
$R_1(4)$
$R_1(3)$
$R_1(2)$
$R_1(1)$

FIG.8

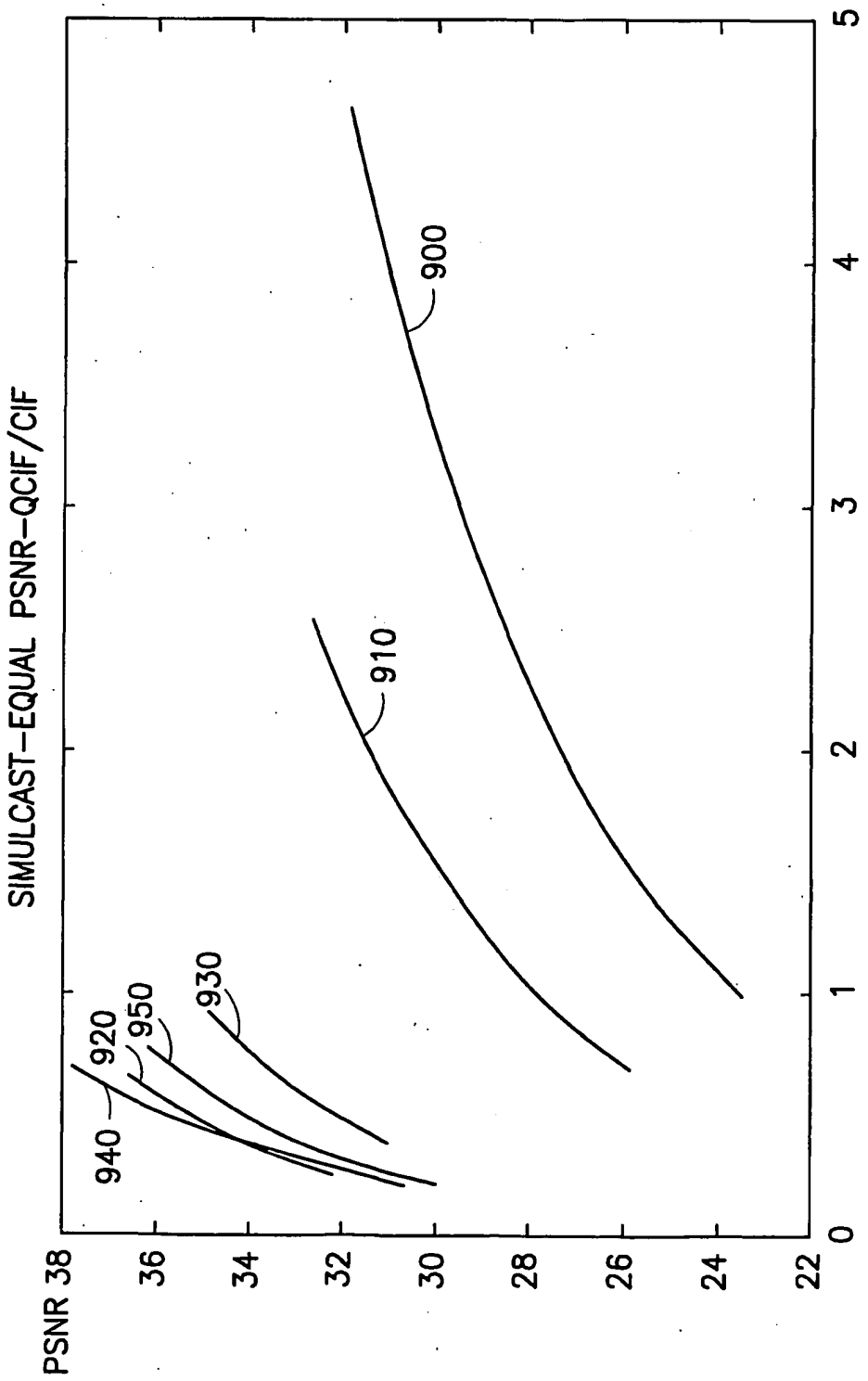

SIMULCAST-EQUAL PSNR-QCIF/CIF
PSNR 38
36
34
32
30
28
26
24
22
0
1
2
3
4
5
TOTAL BITRATE (Mbps)
900
910
930
920
950
940

FIG.9

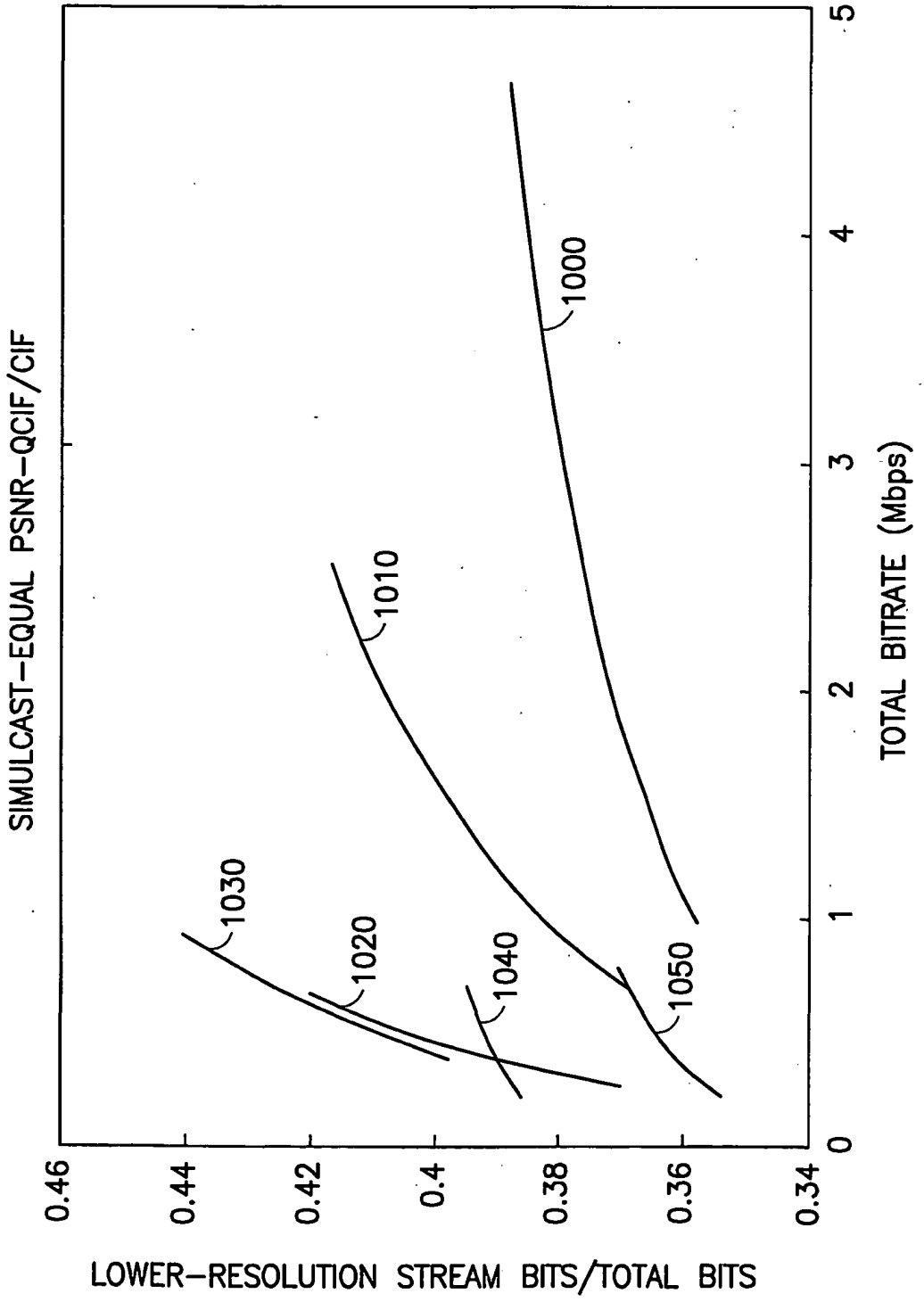

SIMULCAST-EQUAL PSNR-QCIF/CIF
LOWER-RESOLUTION STREAM BITS/TOTAL BITS
TOTAL BITRATE (Mbps)
0.46
0.44
0.42
0.4
0.38
0.36
0.34
0
1
2
3
4
5
1000
1010
1020
1030
1040
1050

FIG.10

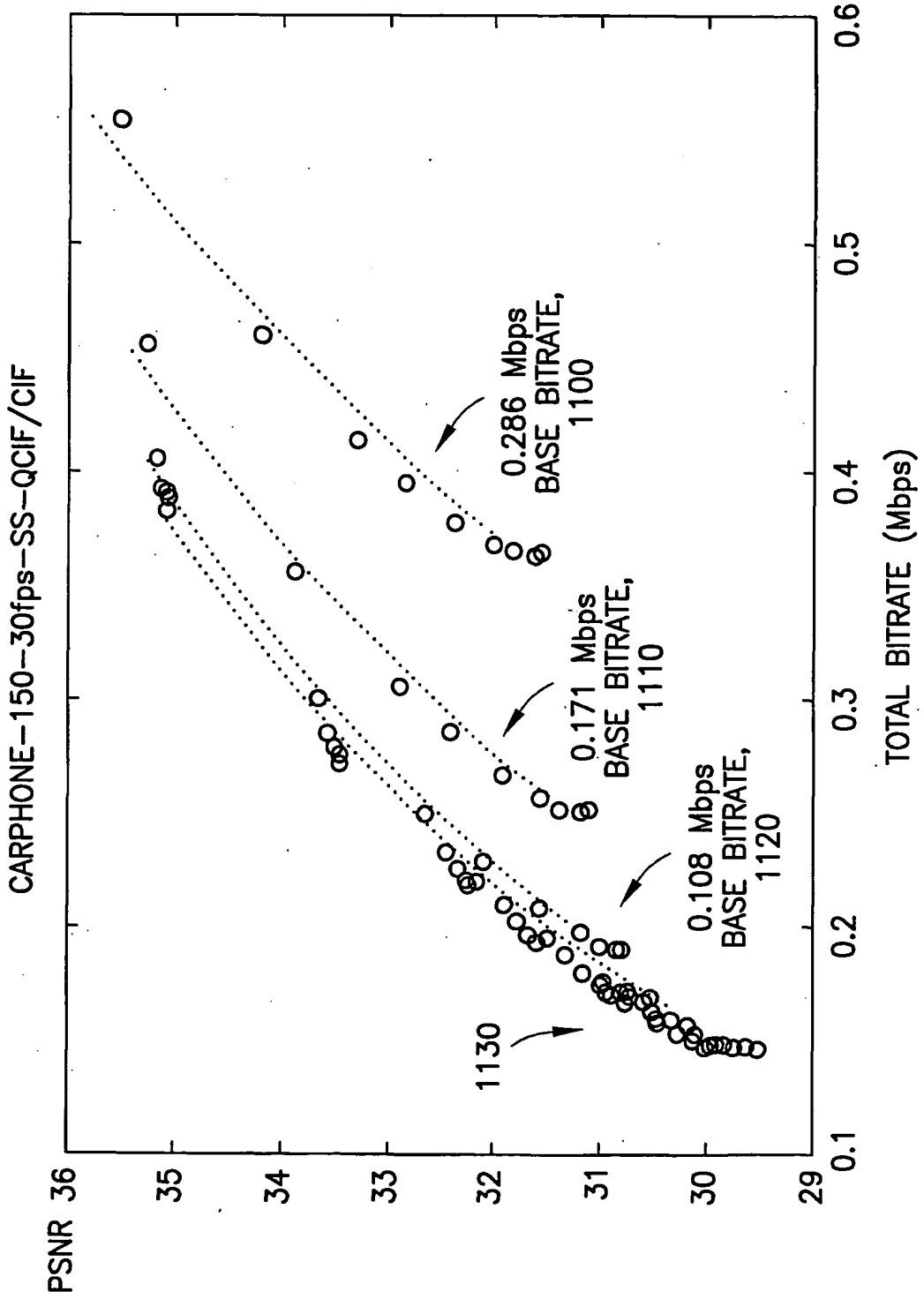

CARPHONE-150-30fps-SS-QCIF/CIF
PSNR 36
35
34
33
32
31
30
29
0.1
0.2
0.3
0.4
0.5
0.6
TOTAL BITRATE (Mbps)
0.286 Mbps
BASE BITRATE,
1100
0.171 Mbps
BASE BITRATE,
1110
0.108 Mbps
BASE BITRATE,
1120
1130

FIG.11

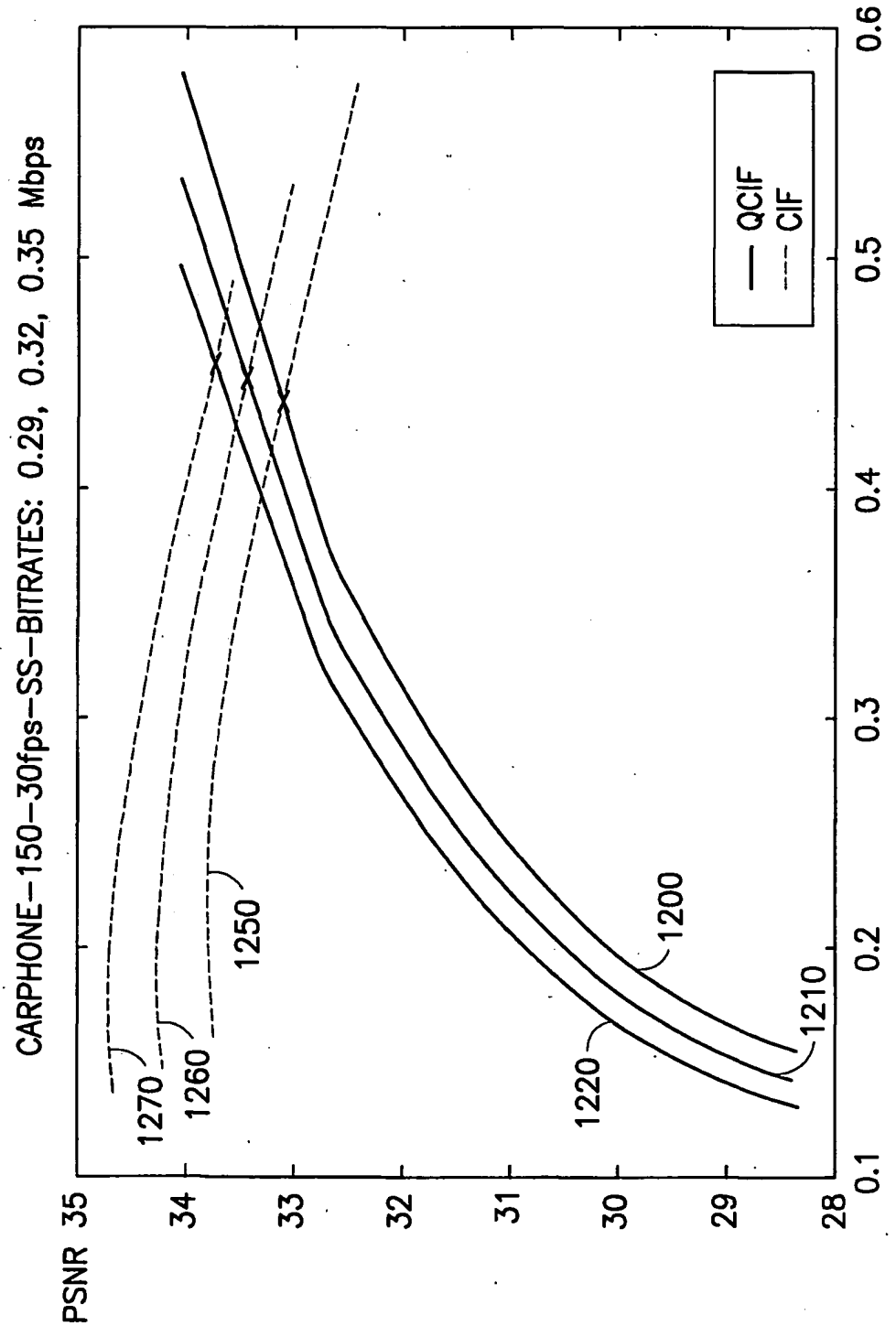

CARPHONE-150-30fps-SS-BITRATES: 0.29, 0.32, 0.35 Mbps
PSNR
35
34
33
32
31
30
29
28
0.1
0.2
0.3
0.4
0.5
0.6
BASE LAYER BITS/TOTAL BITS
QCIF
CIF
1270
1260
1250
1200
1210
1220

FIG.12

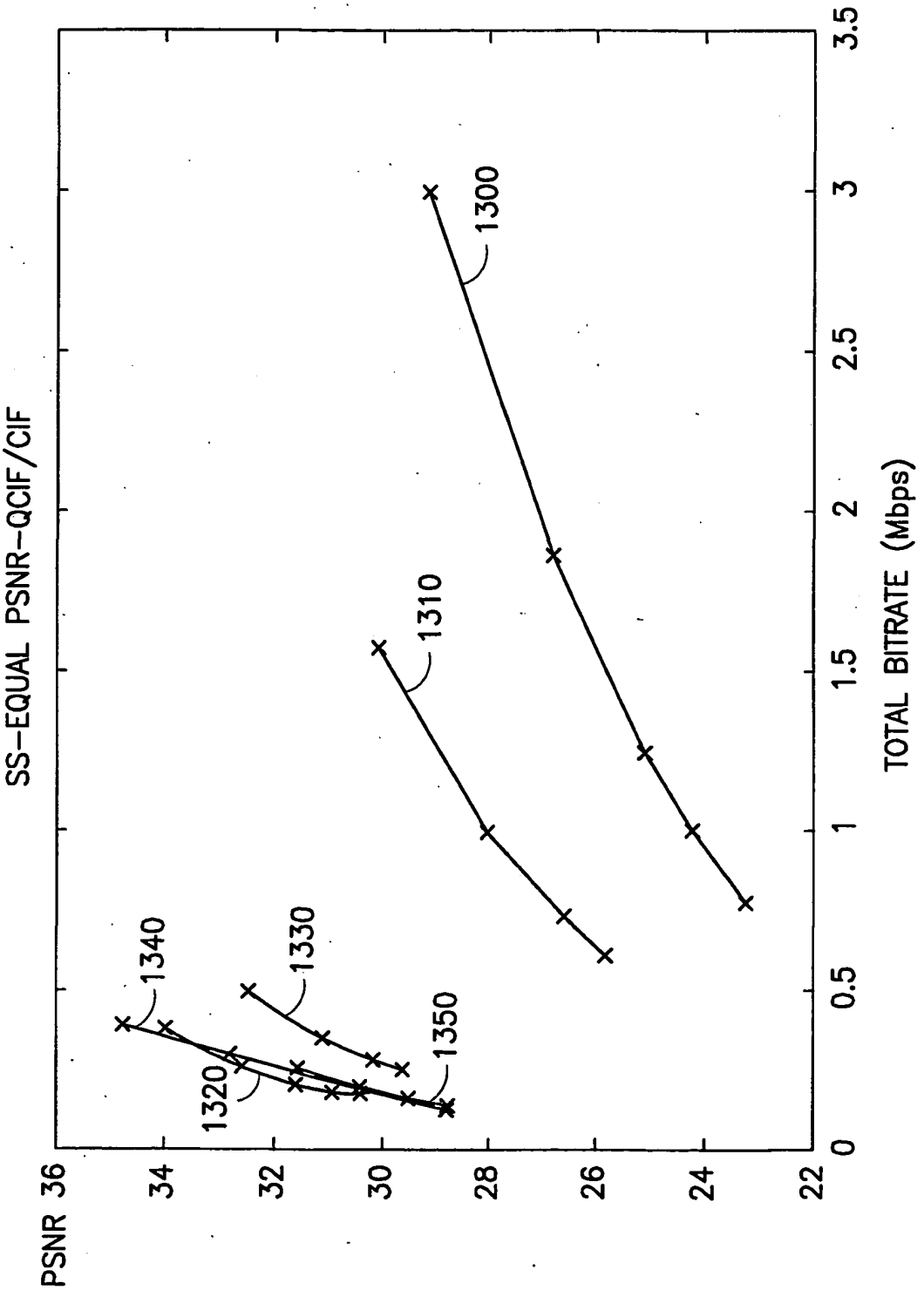

SS-EQUAL PSNR-QCIF/CIF
PSNR 36
34
32
30
28
26
24
22
0
0.5
1
1.5
2
2.5
3
3.5
TOTAL BITRATE (Mbps)
1300
1310
1320
1330
1340
1350

FIG.13

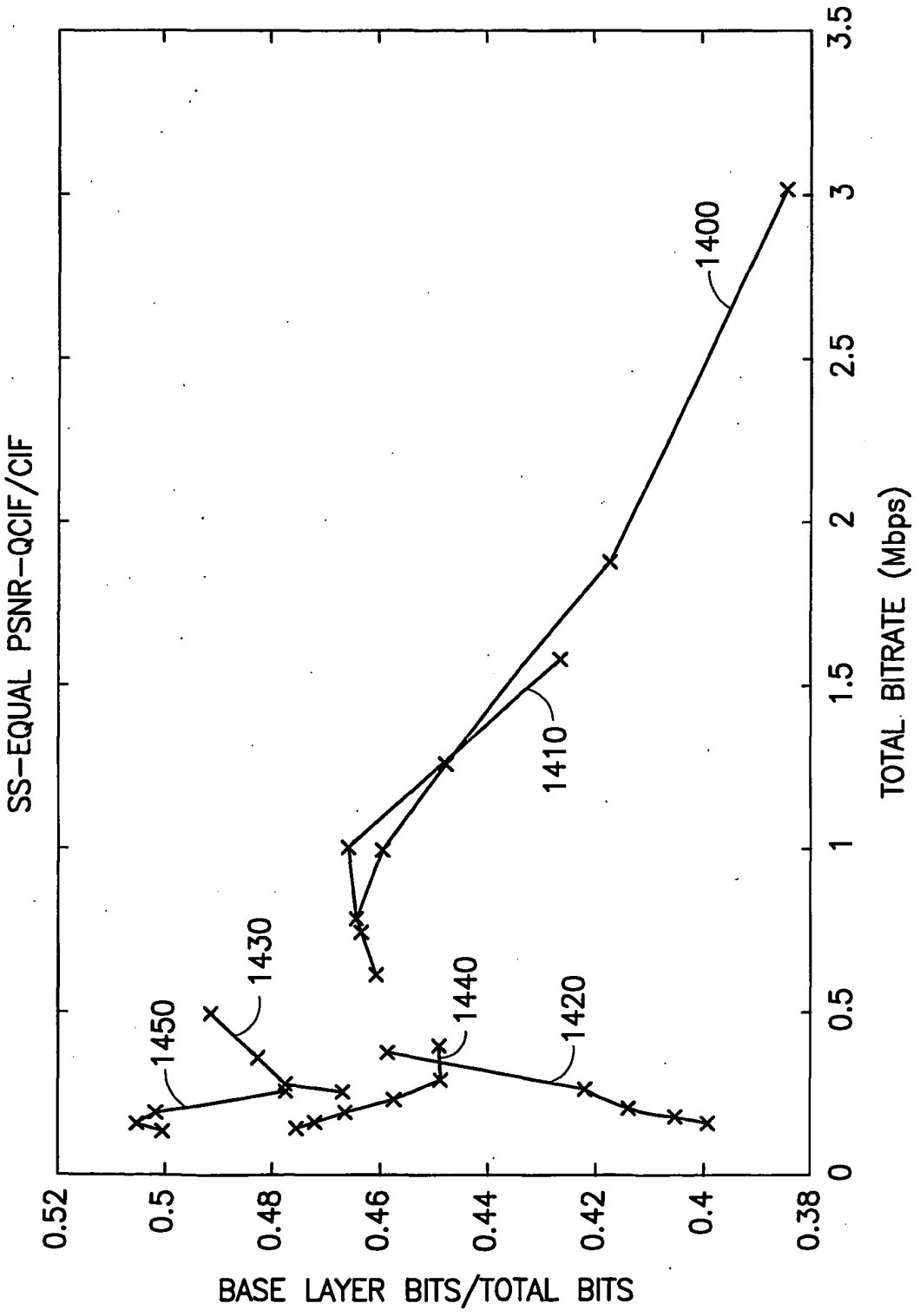

SS-EQUAL PSNR-QCIF/CIF
BASE LAYER BITS/TOTAL BITS
TOTAL BITRATE (Mbps)
0.52
0.5
0.48
0.46
0.44
0.42
0.4
0.38
0
0.5
1
1.5
2
2.5
3
3.5
1400
1410
1420
1430
1440
1450

FIG.14

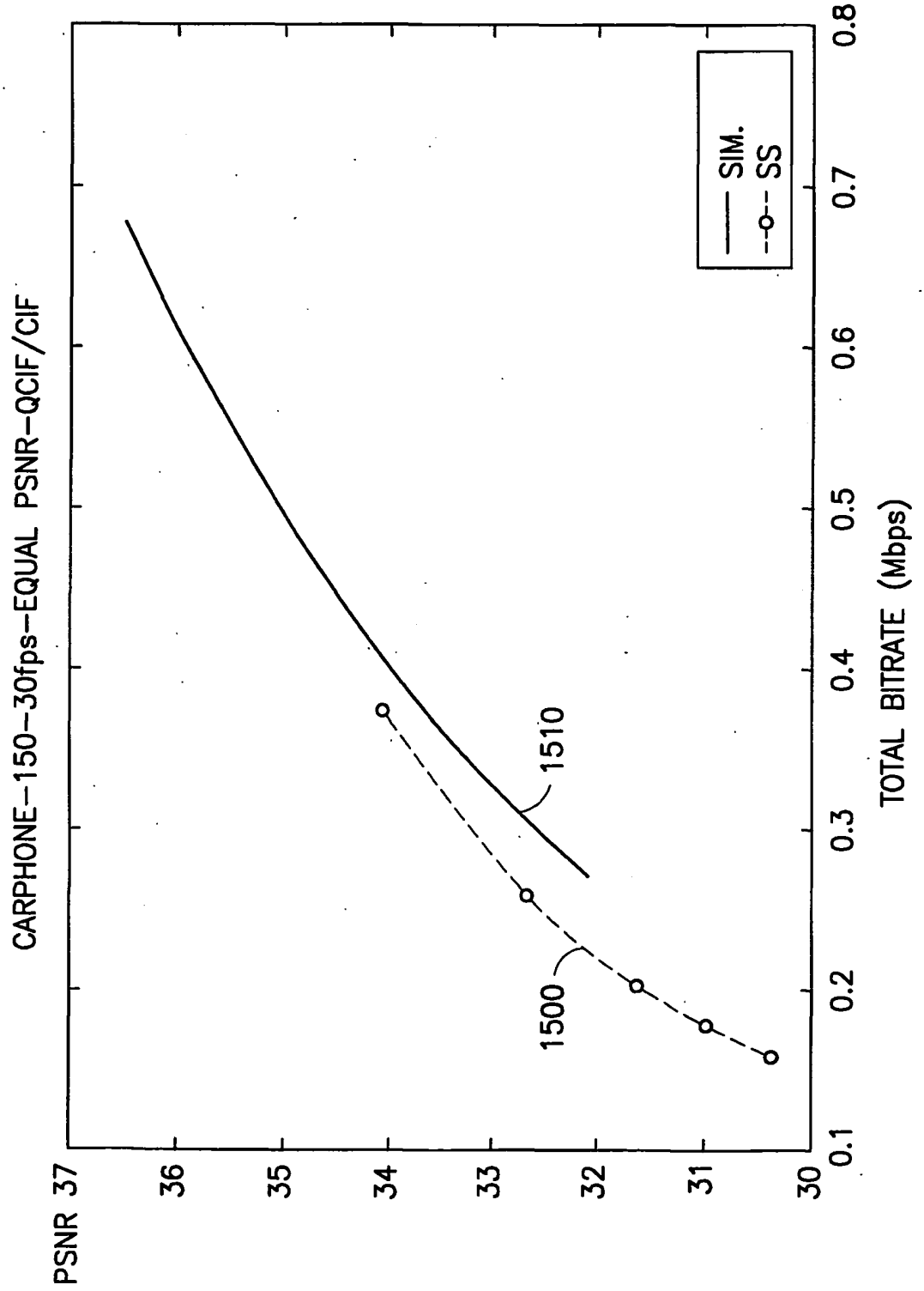

CARPHONE-150-30fps-EQUAL PSNR-QCIF/CIF
PSNR 37
36
35
34
33
32
31
30
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
TOTAL BITRATE (Mbps)
SIM.
SS
1510
1500

FIG.15

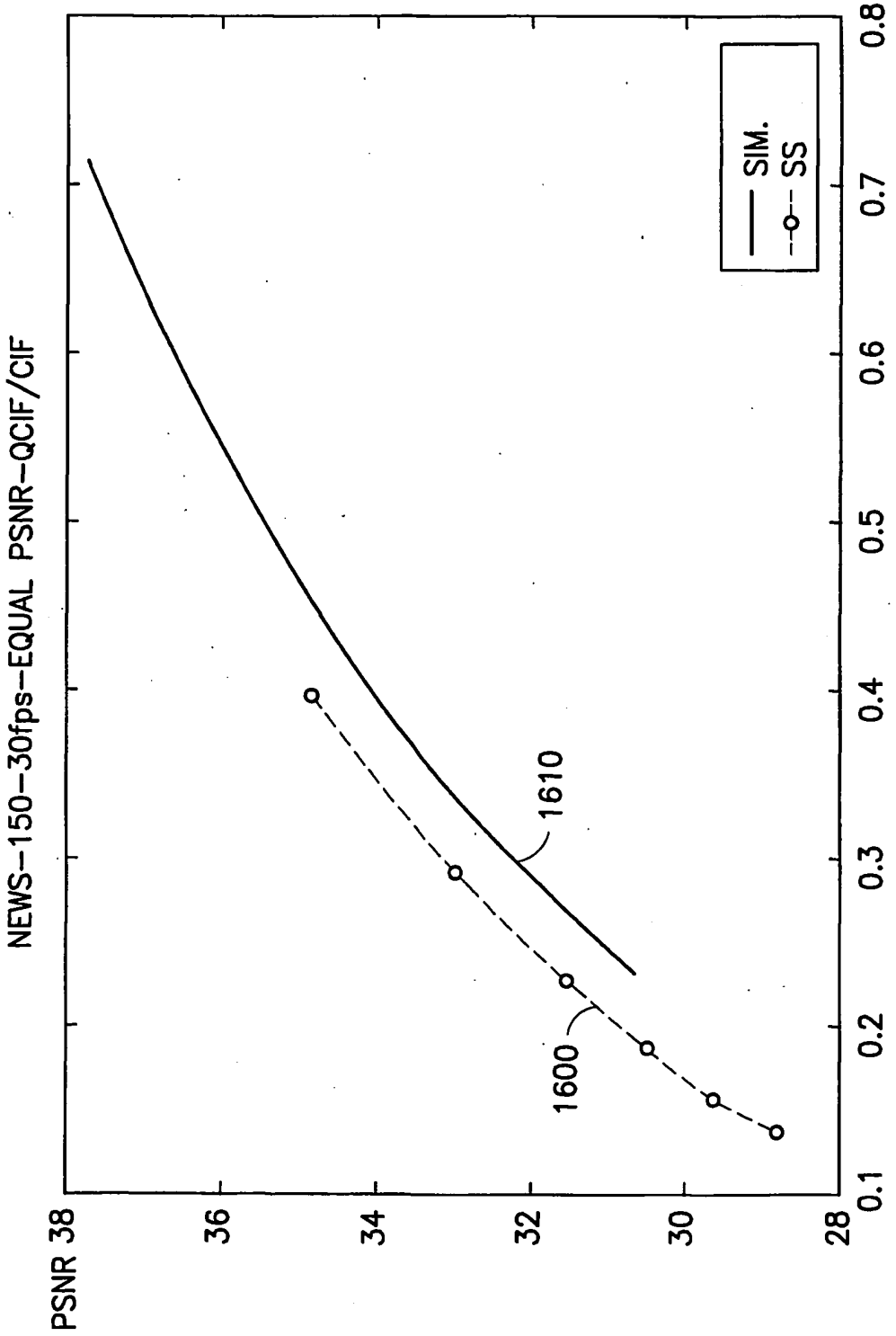

NEWS-150-30fps-EQUAL PSNR-QCIF/CIF
PSNR 38
36
34
32
30
28
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
TOTAL BITRATE (Mbps)
SIM.
SS
1610
1600

FIG.16

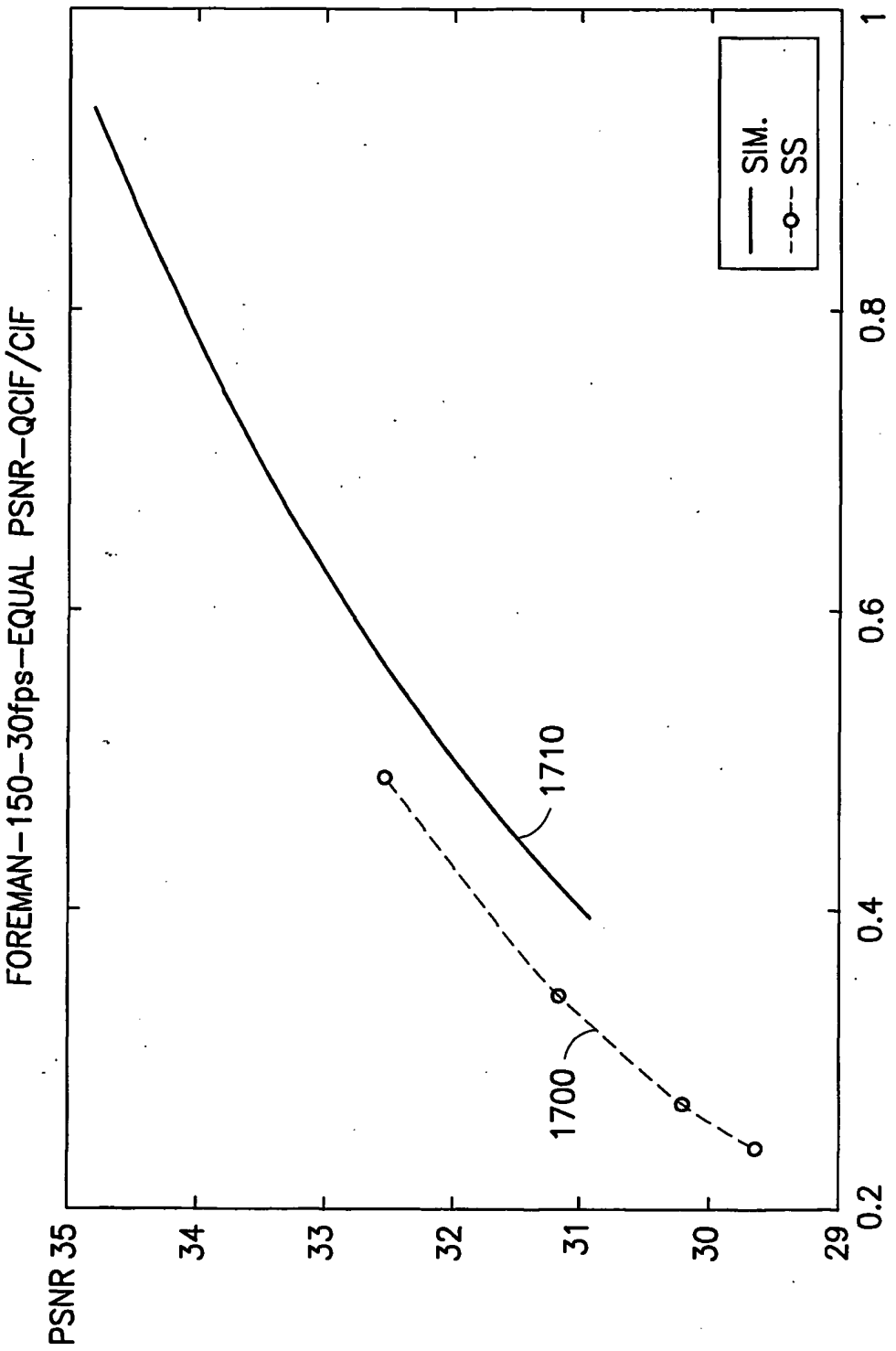

FOREMAN–150–30fps–EQUAL PSNR–QCIF/CIF
PSNR 35
34
33
32
31
30
29
0.2
0.4
0.6
0.8
1
TOTAL BITRATE (Mbps)
SIM.
SS
1710
1700

FIG.17

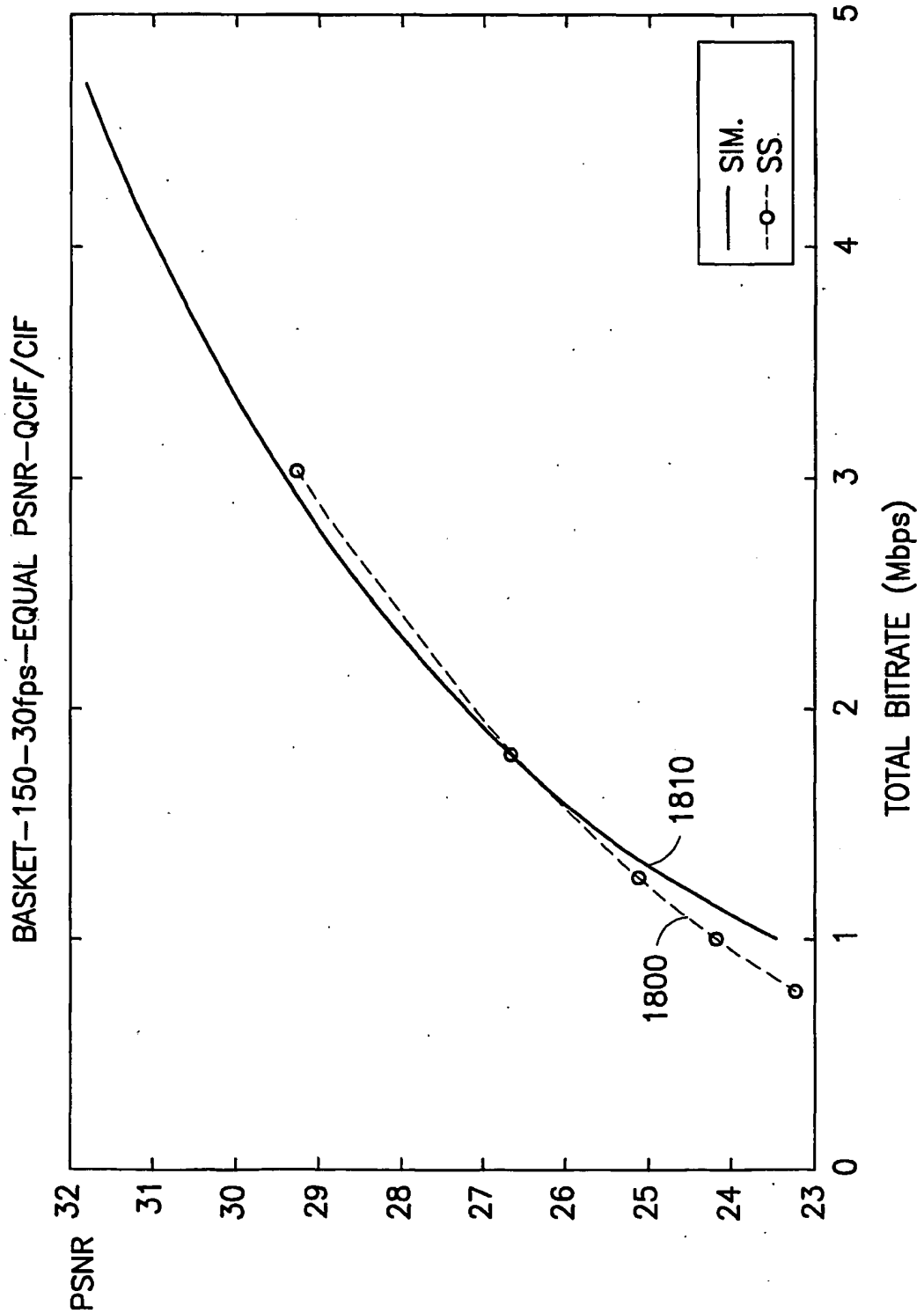

BASKET-150-30fps-EQUAL PSNR-QCIF/CIF
PSNR
32
31
30
29
28
27
26
25
24
23
TOTAL BITRATE (Mbps)
0
1
2
3
4
5
SIM.
SS
1810
1800

FIG.18

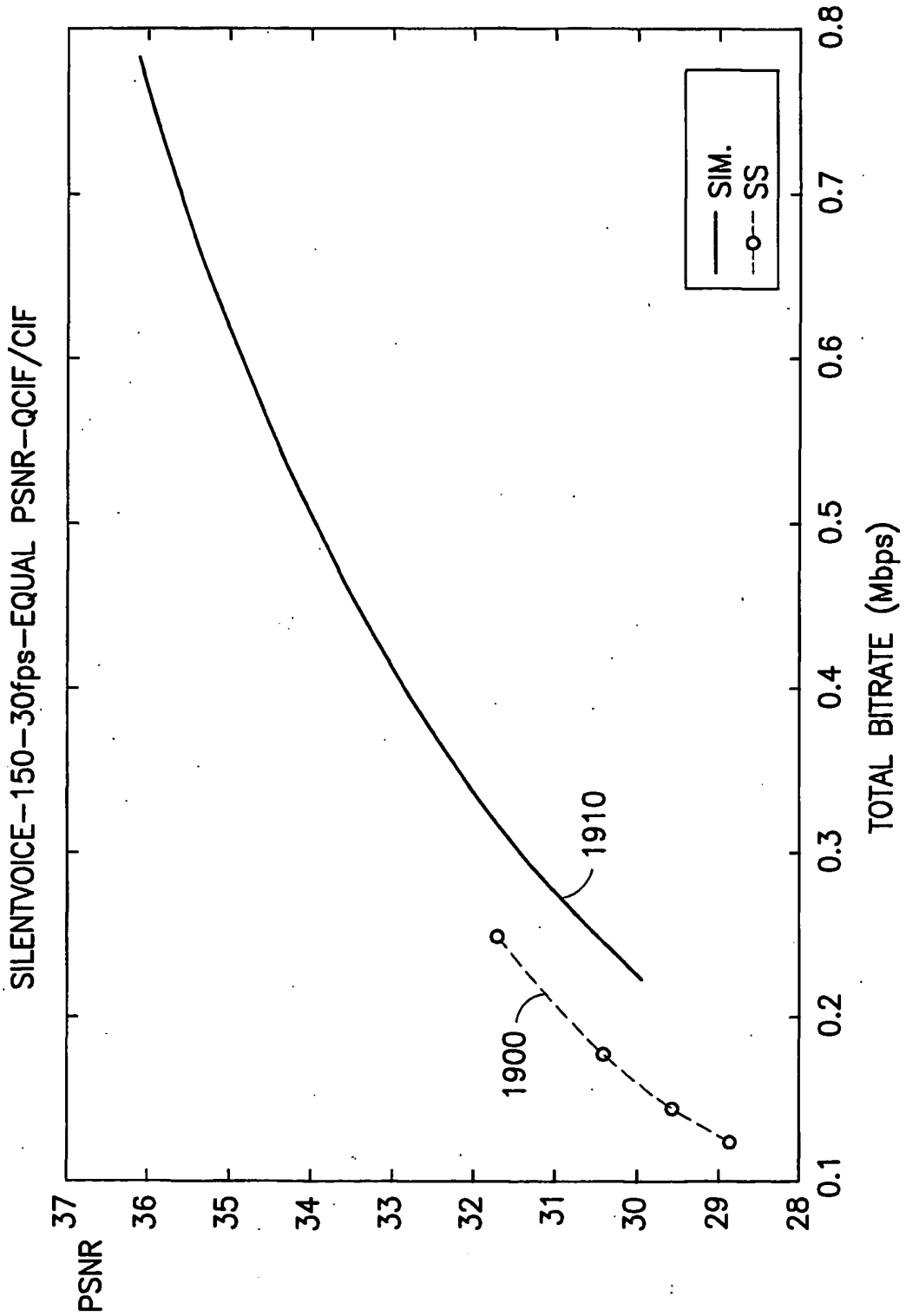

SILENTVOICE-150-30fps-EQUAL PSNR-QCIF/CIF
PSNR
TOTAL BITRATE (Mbps)
SIM.
SS
1910
1900
37
36
35
34
33
32
31
30
29
28
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

FIG.19

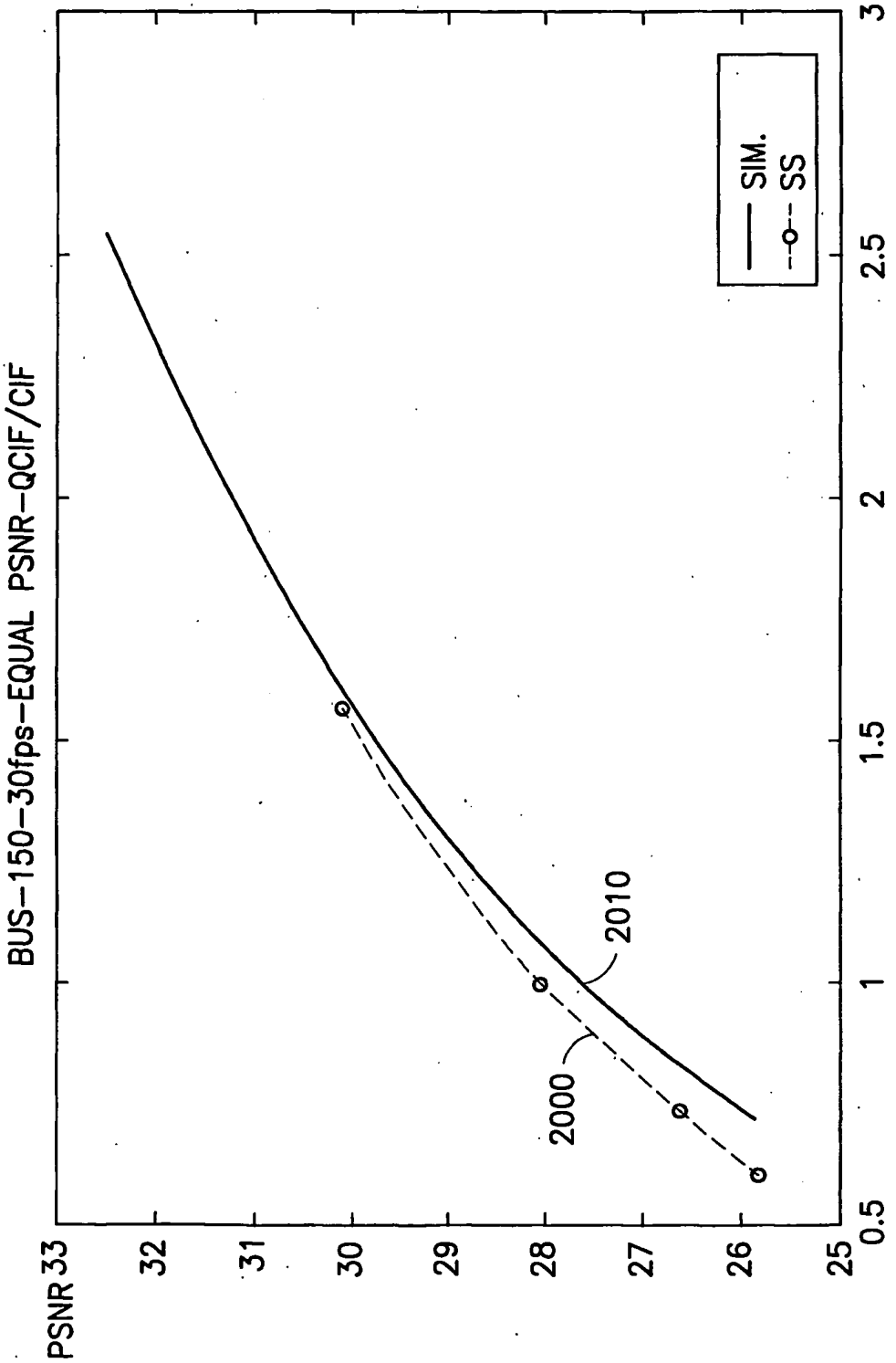

BUS-150-30fps-EQUAL PSNR-QCIF/CIF
PSNR
33
32
31
30
29
28
27
26
25
0.5
1
1.5
2
2.5
3
TOTAL BITRATE (Mbps)
SIM.
SS
2010
2000

FIG.20

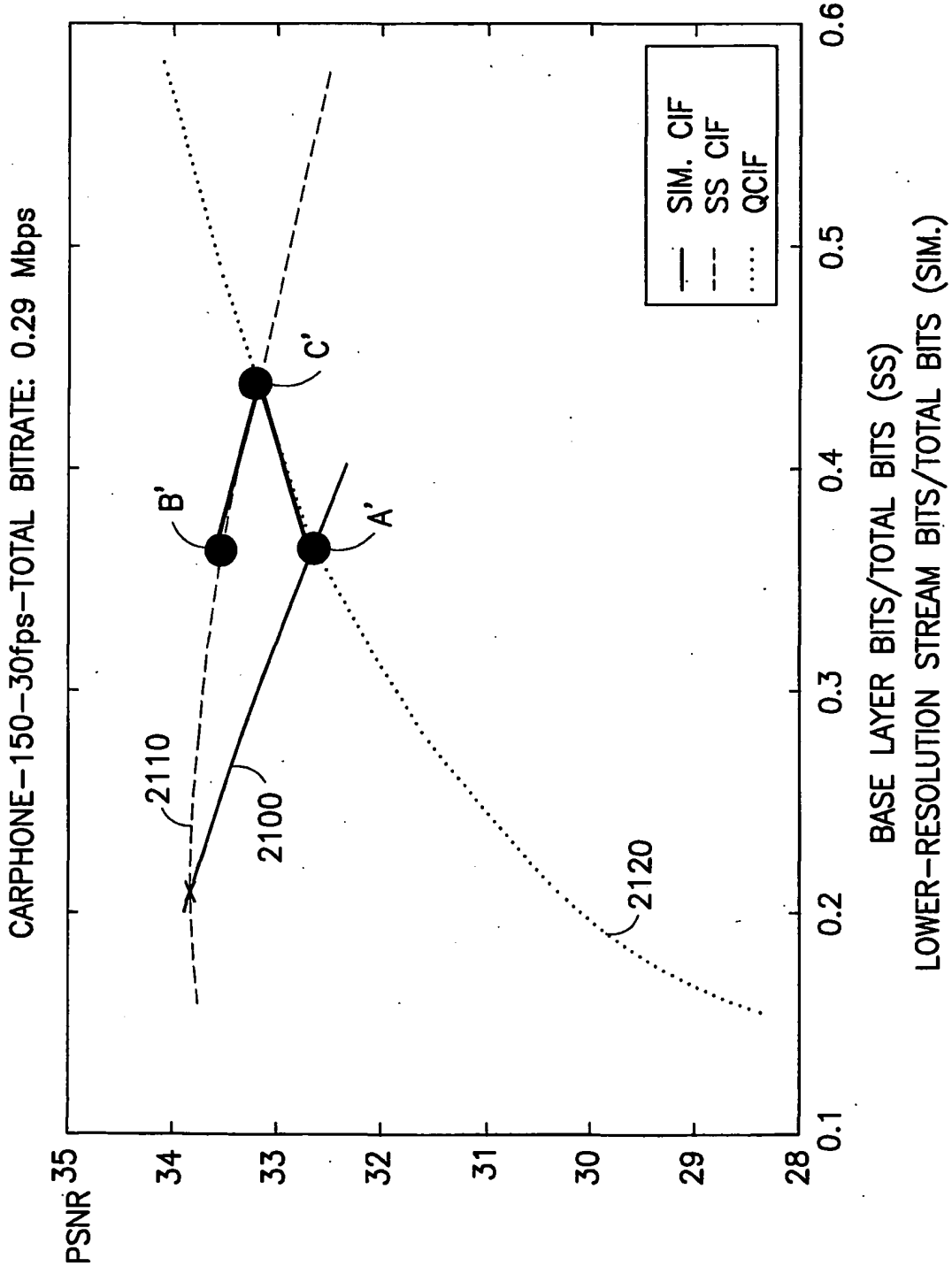

CARPHONE-150-30fps-TOTAL BITRATE: 0.29 Mbps
PSNR
35
34
33
32
31
30
29
28
0.1
0.2
0.3
0.4
0.5
0.6
BASE LAYER BITS/TOTAL BITS (SS)
LOWER-RESOLUTION STREAM BITS/TOTAL BITS (SIM.)
SIM. CIF
SS CIF
QCIF
A'
B'
C'
2110
2100
2120

FIG.21

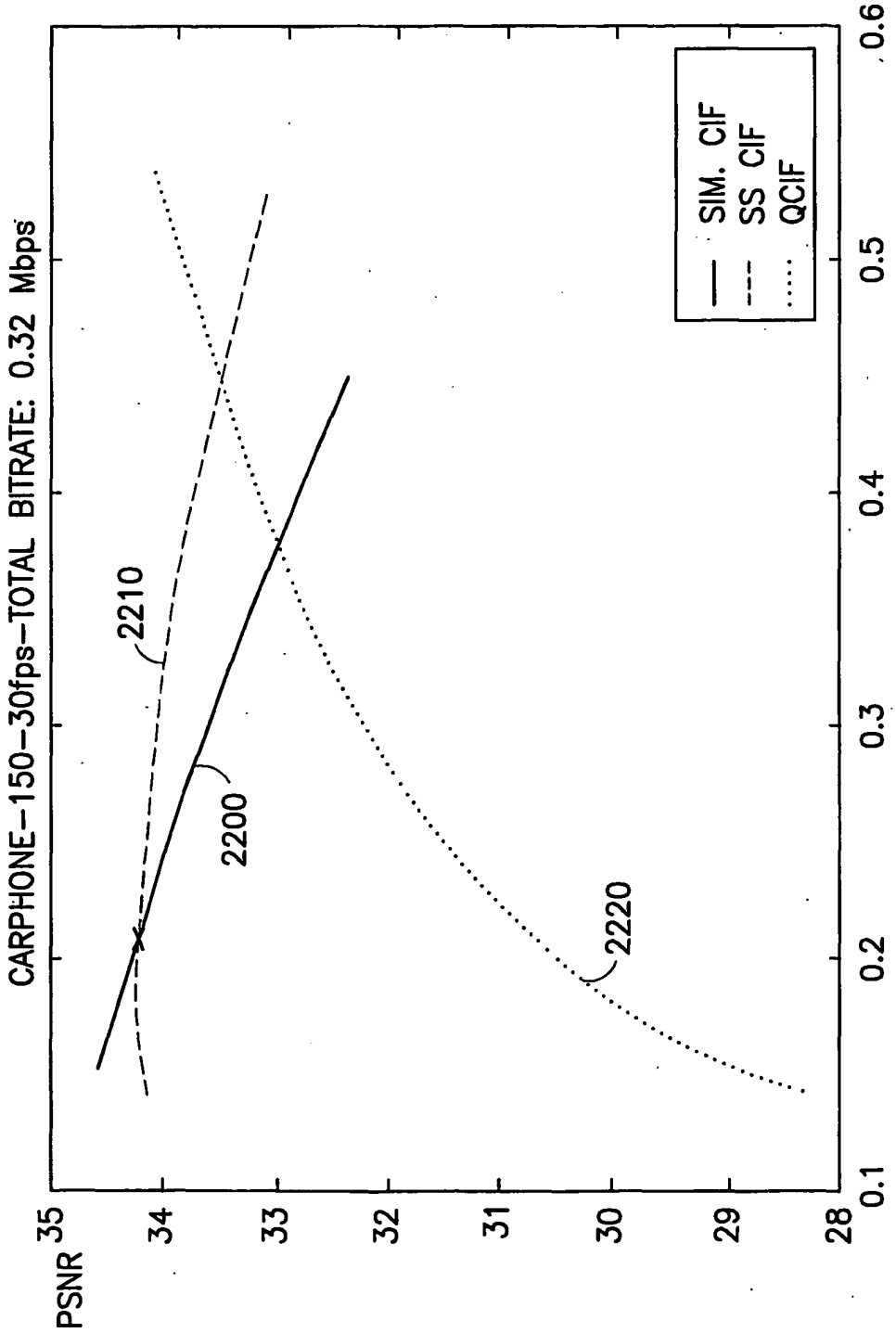

CARPHONE-150-30fps-TOTAL BITRATE: 0.32 Mbps
PSNR
35
34
33
32
31
30
29
28
0.1
0.2
0.3
0.4
0.5
0.6
BASE LAYER BITS/TOTAL BITS (SS)
LOWER-RESOLUTION STREAM BITS/TOTAL BITS (SIM.)
SIM. CIF
SS CIF
QCIF
2210
2200
2220

FIG.22

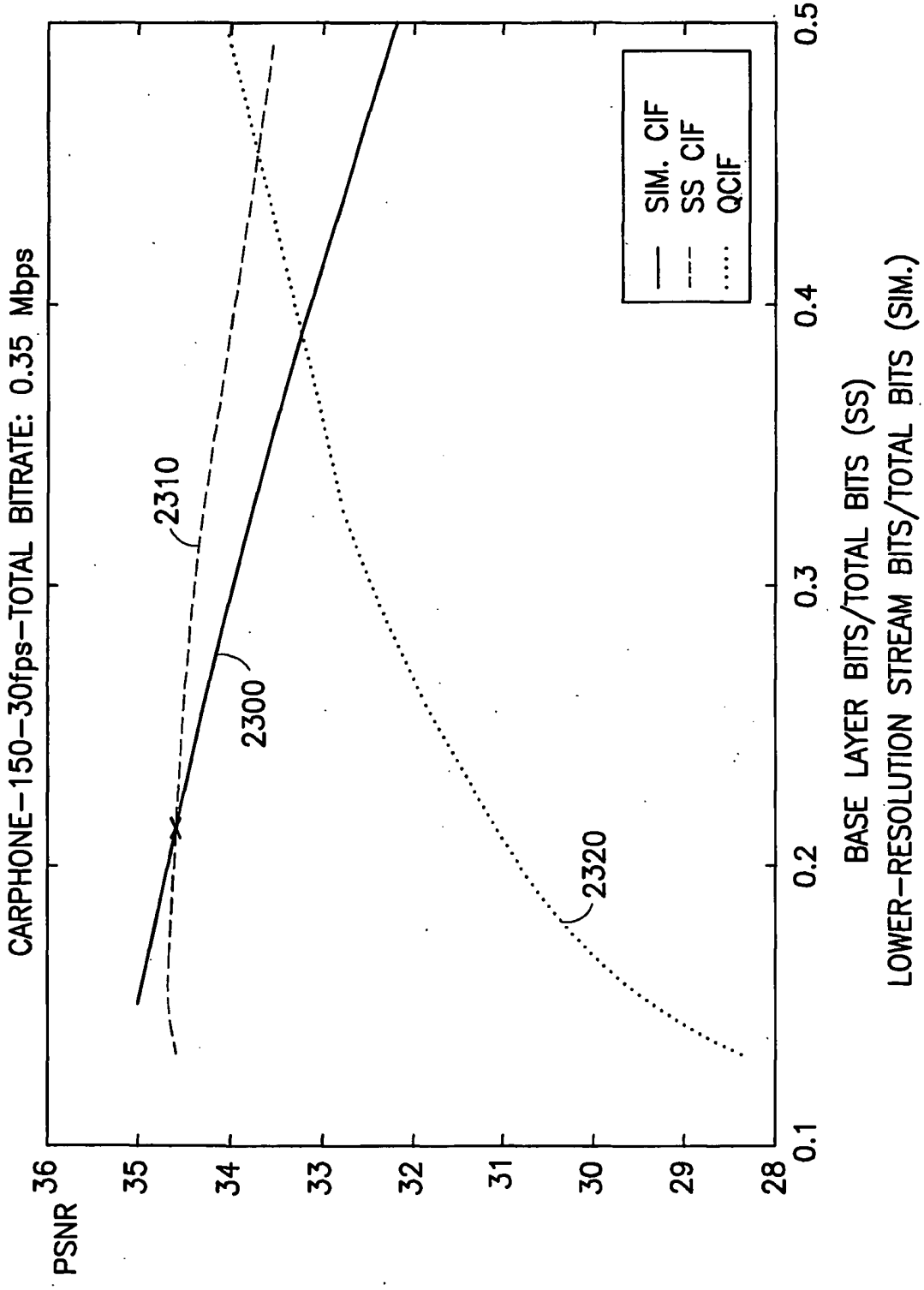

CARPHONE-150-30fps-TOTAL BITRATE: 0.35 Mbps
PSNR
36
35
34
33
32
31
30
29
28
0.1
0.2
0.3
0.4
0.5
BASE LAYER BITS/TOTAL BITS (SS)
LOWER-RESOLUTION STREAM BITS/TOTAL BITS (SIM.)
SIM. CIF
SS CIF
QCIF
2310
2300
2320

FIG.23

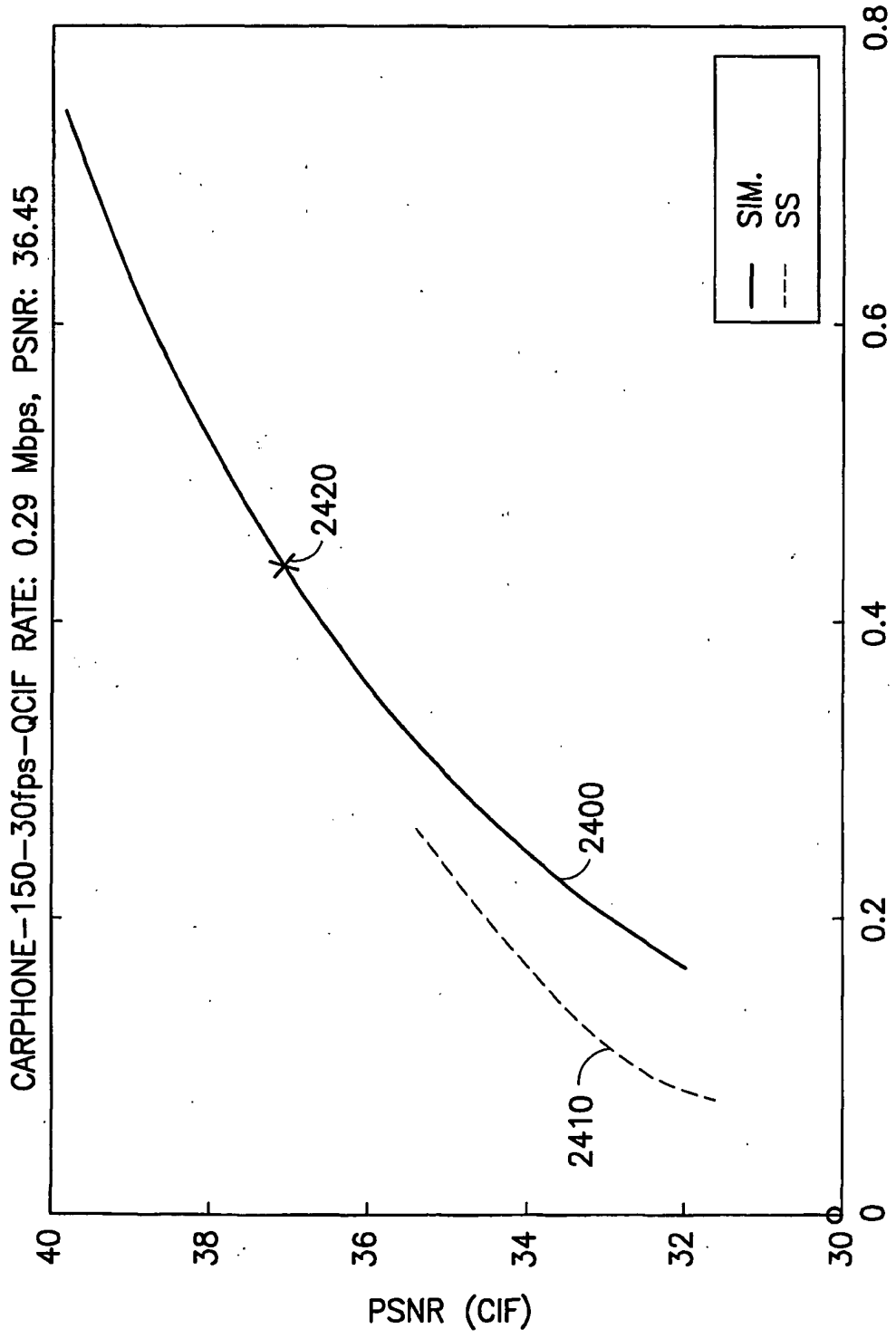

CARPHONE-150-30fps-QCIF RATE: 0.29 Mbps, PSNR: 36.45
SIM.
SS
2420
2400
2410
40
38
36
34
32
30
PSNR (CIF)
0
0.2
0.4
0.6
0.8
ENHANCEMENT LAYER BITRATE (SS) (Mbps)
HIGHER-RESOLUTION STREAM BITRATE (SIM.) (Mbps)

FIG.24

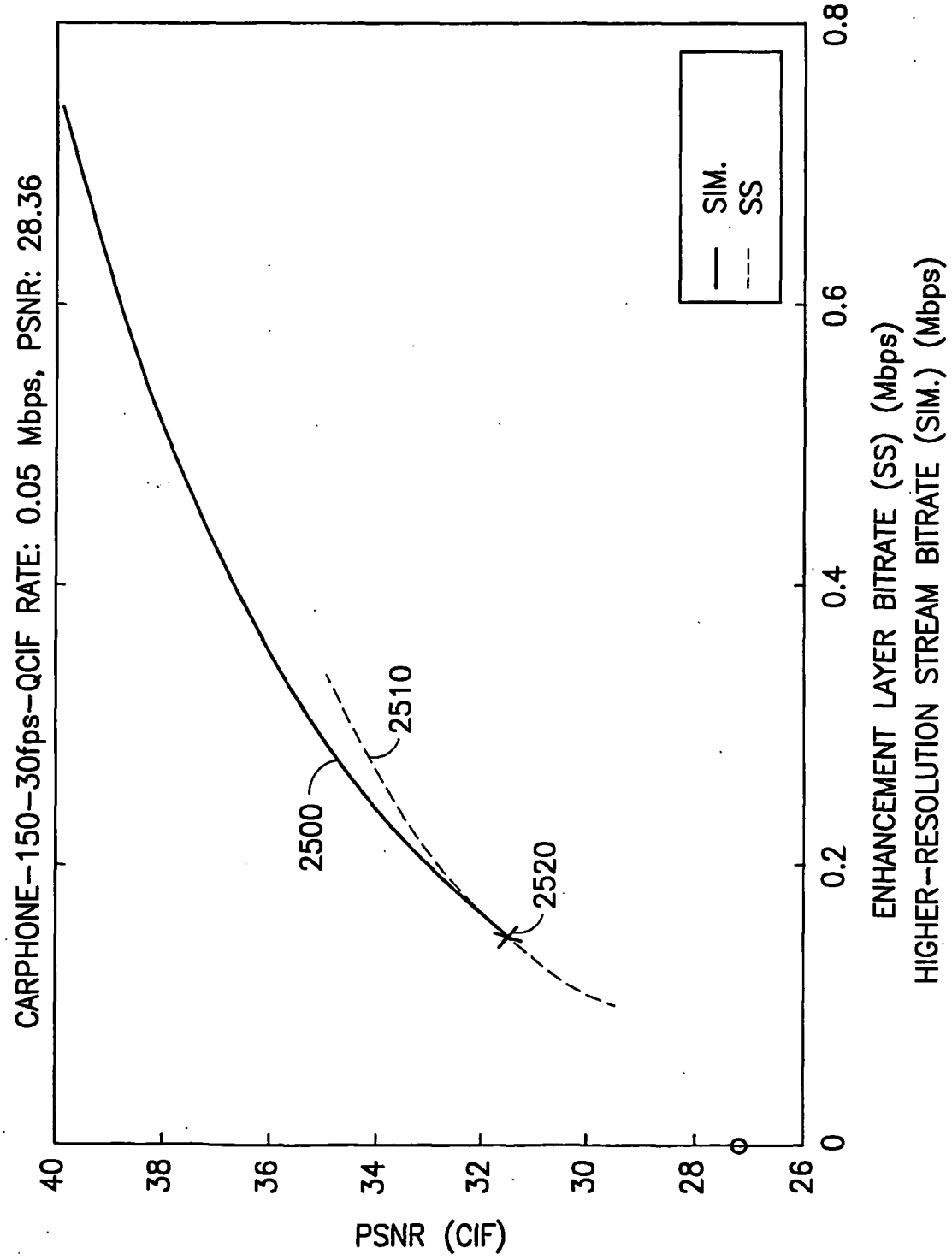

CARPHONE-150-30fps-QCIF RATE: 0.05 Mbps, PSNR: 28.36
SIM.
SS
2500
2510
2520
PSNR (CIF)
40
38
36
34
32
30
28
26
0
0.2
0.4
0.6
0.8
ENHANCEMENT LAYER BITRATE (SS) (Mbps)
HIGHER-RESOLUTION STREAM BITRATE (SIM.) (Mbps)

FIG.25

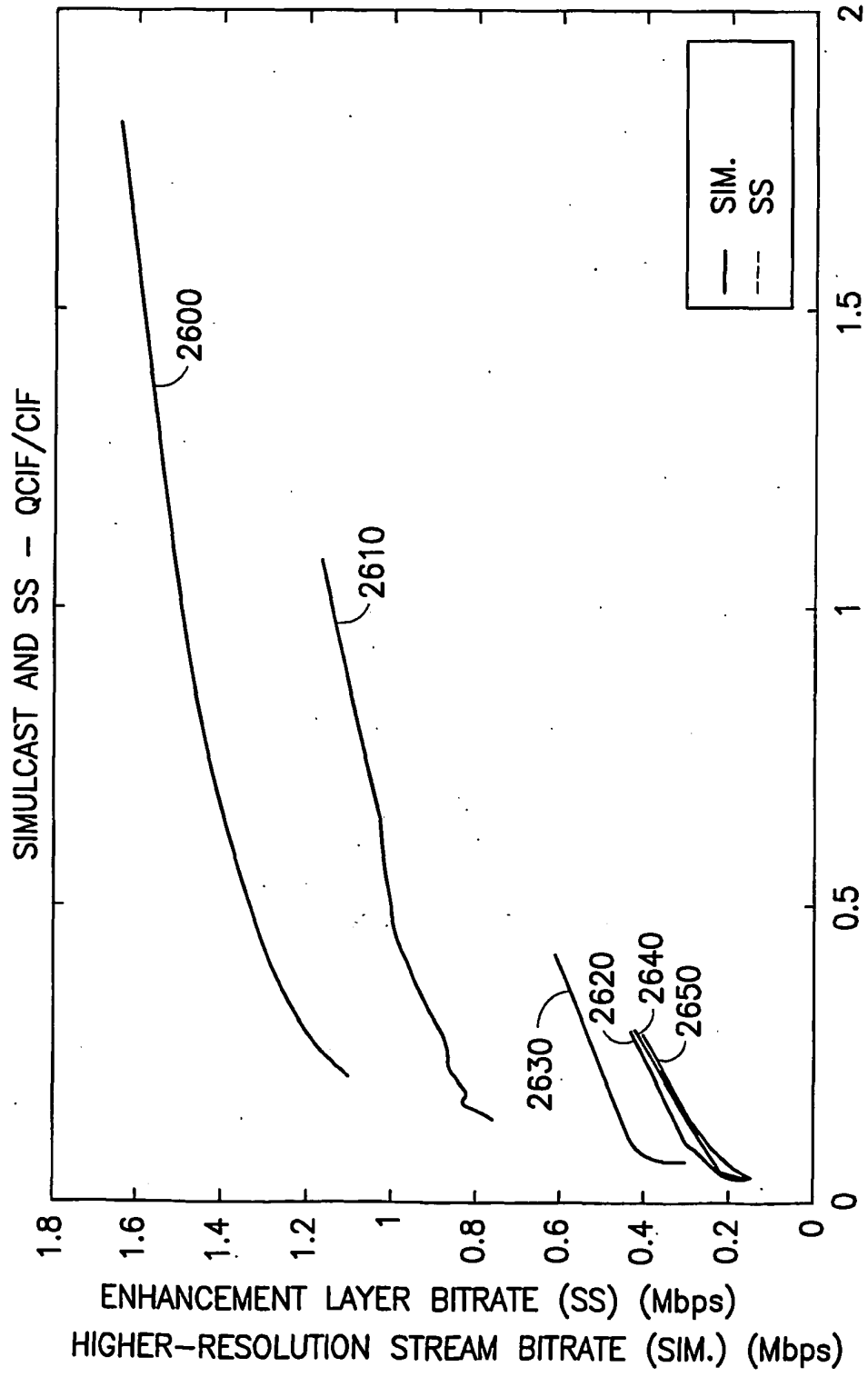

SIMULCAST AND SS – QCIF/CIF
SIM.
SS
2600
2610
2630
2620
2640
2650
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
ENHANCEMENT LAYER BITRATE (SS) (Mbps)
HIGHER–RESOLUTION STREAM BITRATE (SIM.) (Mbps)
BASE LAYER BITRATE (SS) (Mbps)
LOWER–RESOLUTION STREAM BITRATE (SIM.) (Mbps)

FIG.26

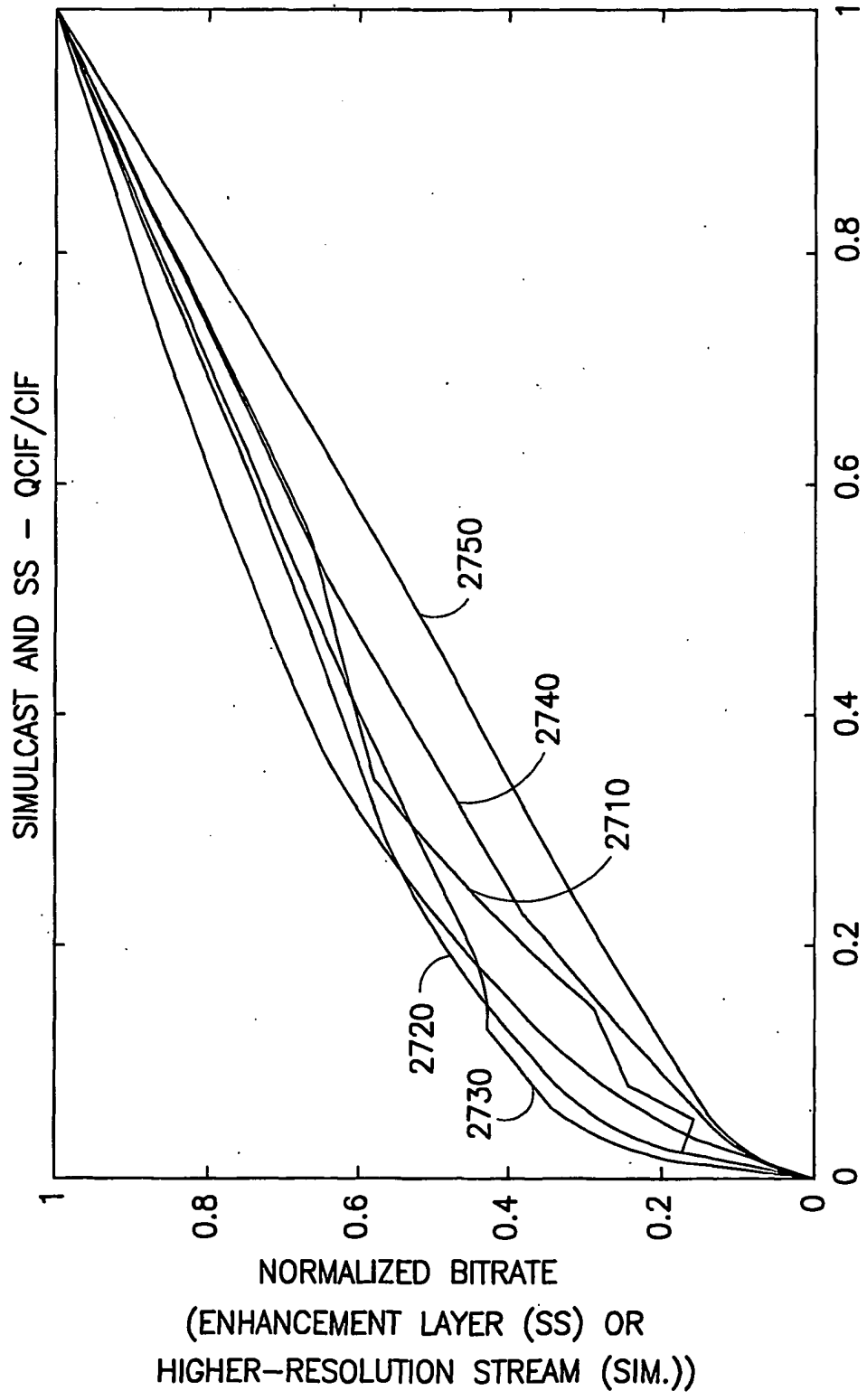

SIMULCAST AND SS – QCIF/CIF
NORMALIZED BITRATE
(ENHANCEMENT LAYER (SS) OR
HIGHER–RESOLUTION STREAM (SIM.))
NORMALIZED BITRATE
(BASE LAYER (SS) OR LOWER–RESOLUTION STREAM (SIM.))
2750
2740
2710
2720
2730
0
0.2
0.4
0.6
0.8
1

FIG.27

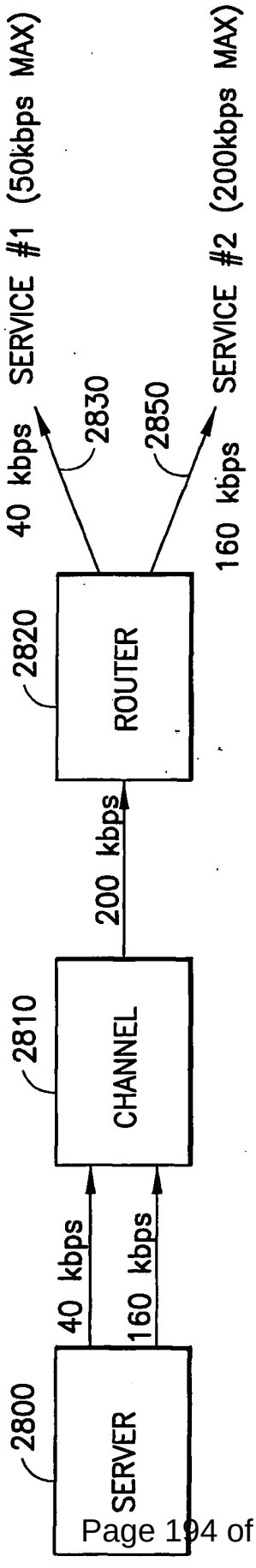

SERVICE #1 (50kbps MAX)
SERVICE #2 (200kbps MAX)
40 kbps
160 kbps
2830
2850
2820
ROUTER
200 kbps
2810
CHANNEL
40 kbps
160 kbps
2800
SERVER

FIG.28

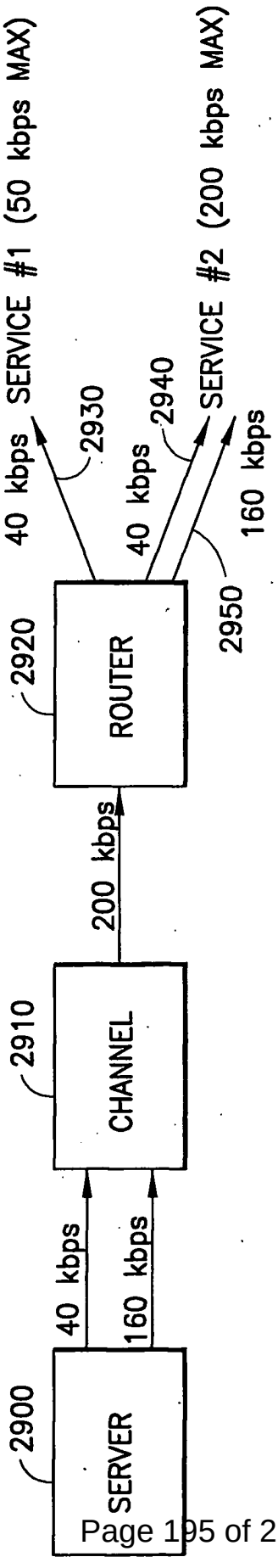

SERVER
2900
CHANNEL
2910
ROUTER
2920
40 kbps
160 kbps
200 kbps
40 kbps
160 kbps
2930
2940
2950
SERVICE #1 (50 kbps MAX)
SERVICE #2 (200 kbps MAX)

FIG.29

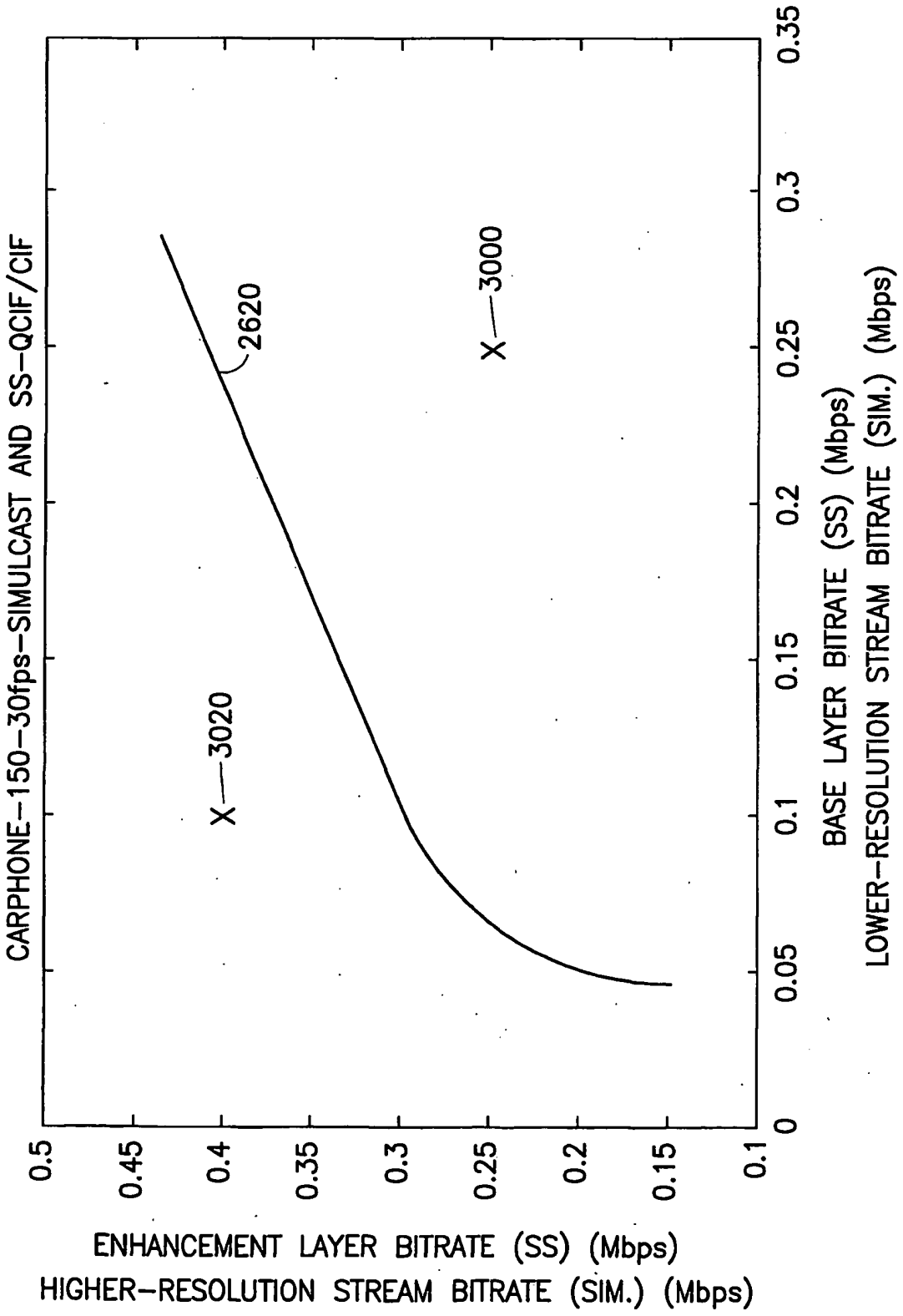

CARPHONE-150-30fps-SIMULCAST AND SS-QCIF/CIF
2620
X—3000
X—3020
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
ENHANCEMENT LAYER BITRATE (SS) (Mbps)
HIGHER-RESOLUTION STREAM BITRATE (SIM.) (Mbps)
BASE LAYER BITRATE (SS) (Mbps)
LOWER-RESOLUTION STREAM BITRATE (SIM.) (Mbps)

FIG.30

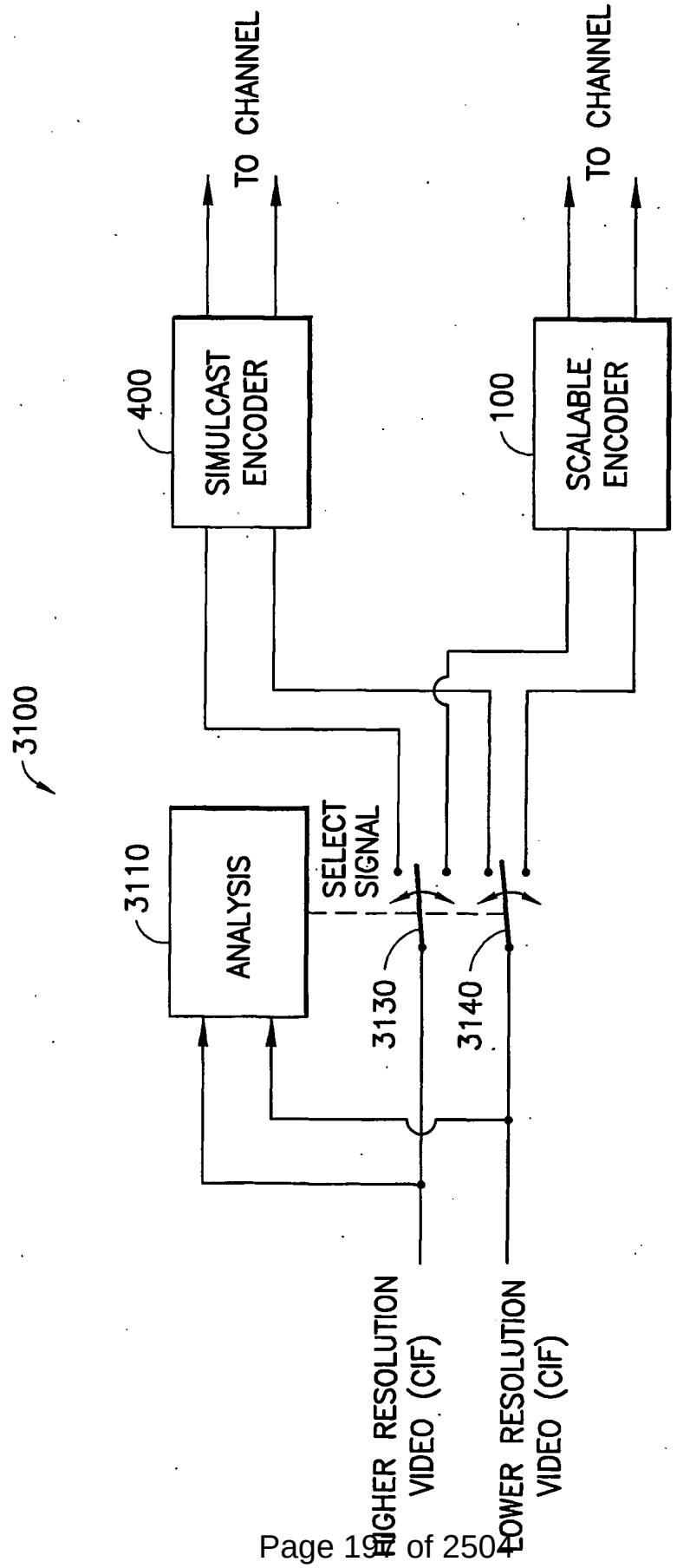

3100
3110
ANALYSIS
SELECT SIGNAL
3130
3140
400
SIMULCAST ENCODER
100
SCALABLE ENCODER
TO CHANNEL
TO CHANNEL
HIGHER RESOLUTION VIDEO (CIF)
LOWER RESOLUTION VIDEO (CIF)

FIG.31

INTERNATIONAL SEARCH REPORT

Int onal Application No
PCT/US 00/09584

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N7/26

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	RADHA H ET AL: "Scalable Internet video using MPEG-4" SIGNAL PROCESSING. IMAGE COMMUNICATION,NL,ELSEVIER SCIENCE PUBLISHERS, AMSTERDAM, vol. 15, no. 1-2, September 1999 (1999-09), pages 95-126, XP004180640 ISSN: 0923-5965	1-6,9, 11,34
A	abstract page 98 -page 100 page 110 -page 111 figures 1,11	30-33,40

-/--

[X] Further documents are listed in the continuation of box C.

[ ] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 September 2000	04/10/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Barel-Faucheux, C

Form PCT/ISA/210 (second sheet) (July 1992)

BEST AVAILABLE COPY

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	ILLGNER K ET AL: "SPATIALLY SCALABLE VIDEO COMPRESSION EMPLOYING RESOLUTION PYRAMIDS" IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS,US,IEEE INC. NEW YORK, vol. 15, no. 9, 1 December 1997 (1997-12-01), pages 1688-1703, XP000726010 ISSN: 0733-8716	12-17,35
A	abstract page 1688 -page 1689 page 1697 figures 6-9	30-33,40
A	CHIANG T ET AL: "HIERARCHICAL HDTV/SDTV COMPATIBLE CODING USING KALMAN STATISTICAL FILTERING" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY,US,IEEE INC. NEW YORK, vol. 9, no. 3, April 1999 (1999-04), pages 424-437, XP000824567 ISSN: 1051-8215 abstract figures 10-17	12-40

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 July 2002 (11.07.2002)

PCT

(10) International Publication Number
WO 02/054776 A1

(51) International Patent Classification[7]: H04N 7/26, 7/58

(21) International Application Number: PCT/FI02/00004

(22) International Filing Date: 3 January 2002 (03.01.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/259,529	3 January 2001 (03.01.2001)	US
09/827,796	6 April 2001 (06.04.2001)	US
09/883,887	18 June 2001 (18.06.2001)	US
09/925,769	9 August 2001 (09.08.2001)	US

(71) Applicant *(for all designated States except US)*: **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **KARCZEWICZ, Marta** [PL/US]; 1224 Hidden Ridge, #3097, Irving, TX 75038 (US). **KURCEREN, Ragip** [TR/US]; 7914 N. MacArthur Blvd., Apt. 2043, Irving, TX 75063 (US).

(74) Agent: **TAMPEREEN PATENTTITOIMISTO OY**; Hermiankatu 6, FIN-33720 Tampere (FI).

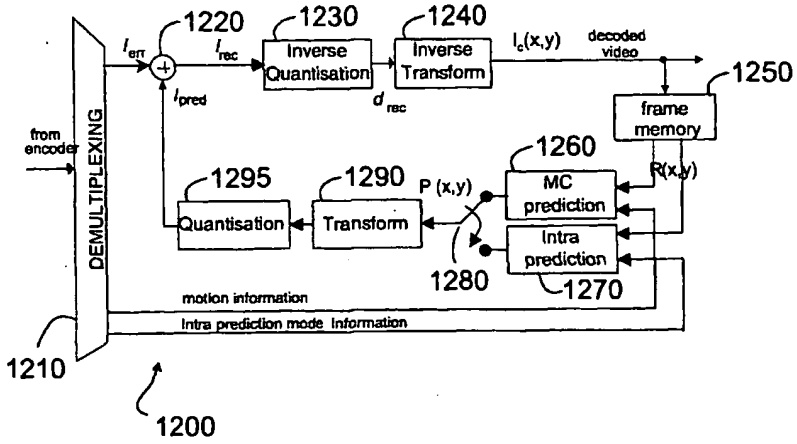
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EC, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: SWITCHING BETWEEN BIT-STREAMS IN VIDEO TRANSMISSION



(57) Abstract: The invention relates to a method for transmitting video information, in which at least a first bit-stream (510) and a second bit-stream are formed. The first bit-stream (510) comprises at least one video frame, and the second bit-stream (520) comprises at least one predictive video frame (524). At least partly different encoding parameters are used with encoding of the frames of said first bit-stream (510) and said second bit-stream (520). At least one frame of said first bit-stream (510) is being transmitted, and the transmission is switched over from said first (510) to said second bit-stream (520). In switching the transmission over from said first (510) to said second bit-stream (520), a secondary switching frame (550) is transmitted, which have been encoded using the encoding parameters of the second bit-stream (520) and at least one reference frame from the first bit-stream (510). Said secondary switching frame (550) is used as a reference frame in the reconstruction of said at least one predictive video frame (524) of the second set of video frames. The invention also relates to an encoder for encoding video information, a decoder for decoding a video information, and a signal representing encoded video information.

WO 02/054776 A1

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SWITCHING BETWEEN BIT-STREAMS IN VIDEO TRANSMISSION

The present invention relates to a method for transmitting video information, in which at least a first bit-stream and a second bit-stream are formed from the video information, the first bit-stream comprises a first set of frames comprising at least one video frame, and the second bit-stream comprises a second set of frames comprising at least one predictive video frame, at least partly different encoding parameters are used with encoding of the frames of said first bit-stream and said second bit-stream, at least one frame of said first bit-stream is being transmitted, and the transmission is switched over from said first bit-stream to said second bit-stream. The invention also relates to an encoder comprising means for forming at least a first bit-stream and a second bit-stream from a video information, the first bit-stream comprises a first set of frames comprising at lest one video frame, and the second bit-stream comprises a second set of frames comprising at least one predictive video frame, and means for using at least partly different encoding parameters with encoding of the frames of said first bit-stream and said second bit-stream, means for transmitting at least one frame of said first bit-stream, and means for switching the transmission over from said first bit-stream to said second bit-stream. The invention further relates to a decoder for decoding a video information from a signal comprising frames from at least a first bit-stream and a second bit-stream, which are formed from the video information, the first bit-stream comprises a first set of frames comprising at lest one video frame, and the second bit-stream comprises a second set of frames comprising at least one predictive video frame, and at least partly different encoding parameters are used with encoding of the frames of said first bit-stream and said second bit-stream. The invention further relates to a signal representing encoded video information, and comprising frames from at least a first bit-stream and a second bit-stream, which are formed from the video information, the first bit-stream comprises a first set of frames comprising at lest one video frame, and the second bit-stream comprises a second set of frames comprising at least one predictive video frame, at least partly

CONFIRMATION COPY

different encoding parameters are used with encoding of the frames of said first bit-stream and said second bit-stream.

BACKGROUND

Recently, multimedia applications that include streaming audio and video information have come into greater use. Several international standardization organizations have established and proposed standards for compressing/encoding and decompressing/decoding the audio and video information. MPEG standards, established by the Motion Picture Expert Group, are the most widely accepted international standards in the field of the multimedia applications. VCEG is the "Video Coding Experts Group", working under the direction of the ITU Telecommunication Standardization Sector (ITU-T). This group works on the standard H.26L for the coding of moving pictures.

A typical video stream comprises a sequence of pictures, often referred to as frames. The frames comprise pixels arranged into a rectangular form. In existing video coding standards, such as H.261, H.262, H.263, H.26L and MPEG-4, three main types of pictures are defined: Intra frames (I-frames), Predictive frames (P-frames) and Bi-directional frames (B-frames). Each picture type exploits a different type of redundancy in a sequence of images and consequently results in different level of compression efficiency and, as explained in the following, provides different functionality within the encoded video sequence. An intra frame is a frame of video data that is coded by exploiting only the spatial correlation of the pixels within the frame itself without using any information from the past or the future frames. Intra frames are used as the basis for decoding/decompression of other frames and provide access points to the coded sequence where decoding can begin.

A predictive frame is a frame that is encoded/compressed using motion compensated prediction from a so-called reference frame, *i.e.* one or more previous/subsequent Intra frames or Predictive frames available in an encoder or in a decoder. A bi-directional frame is a frame that is

encoded/compressed by prediction from a previous Intra frame or Predictive frame and/or a subsequent Intra frame or Predictive frame.

Since adjacent frames in a typical video sequence are highly correlated, higher compression can be achieved when using Bi-directional or Predictive frames instead of Intra frames. On the other hand, when the temporal predictive coding is employed within the coded video stream, B-frames and/or P-frames cannot be decoded without correctly decoding all the other previous and/or subsequent reference frames which were used with coding of the Bi-directional and Predictive frames. In situations in which the reference frame(s) used in the encoder and respective reference frame(s) in the decoder are not identical either due to errors during transmission or due to some intentional action on the transmitting side, the subsequent frames that make use of prediction from such a reference frame can not be reconstructed on the decoding side to yield a decoded frame identical to that originally encoded on the encoding side. This mismatch is not only confined to a single frame but further propagates in time due to the use of motion compensated coding.

Figs. 1A-1C illustrate the types of encoded/compressed video frames used in a typical video encoding/decoding system. Advantageously, prior to encoding, the pictures of the video sequence are represented by these matrices of multiple-bit numbers, one representing the luminance (brightness) of the image pixels, and the other two each representing a respective one of two chrominance (colour) components. Fig. 1A depicts the way in which an Intra frame 200 is encoded using only image information present in the frame itself. Fig. 1B illustrates construction of a Predictive frame 210. Arrow 205a represents the use of motion compensated prediction to create the P-frame 210. Fig. 1C depicts construction of Bi-directional frames 220. B-frames are usually inserted between I-frames or P-frames. Fig. 2 represents a group of pictures in display order and illustrates how B-frames inserted between I-and P-frames, as well as showing the direction in which motion compensation information flows. In Figures 1B, 1C and 2, arrows 205a depict forward motion compensation

prediction information necessary to reconstruct P-frames 210, whereas arrows 215a and 215b depict motion compensation information used in reconstructing B-frames 220 in forward direction (215a) and backward direction (215b). In other words, the arrows 205a and 215a show the flow of information when predictive frames are predicted from frames that are earlier in display order than the frame being reconstructed, and arrows 215b show the flow of information when predictive frames are predicted from frames that are later in display order than the frame being reconstructed.

In motion compensated prediction, the similarity between successive frames in a video sequence is utilized to improve coding efficiency. More specifically, so-called motion vectors are used to describe the way in which pixels or regions of pixels move between successive frames of the sequence. The motion vectors provide offset values and error data that refer to a past or a future frame of video data having decoded pixel values that may be used with the error data to compress/encode or decompress/decode a given frame of video data.

The capability to decode/decompress P-frames requires the availability of the previous I- or P-reference frame, furthermore in order to decode a B-frame requires the availability of the subsequent I- or P-reference frame is also required. For example, if an encoded/compressed data stream has the following frame sequence or display order:

$I_1\ B_2\ B_3\ P_4\ B_5\ P_6\ B_7\ P_8\ B_9\ B_{10}\ P_{11} \ldots P_{n-3}\ B_{n-2}\ P_{n-1}\ I_n,$

the corresponding decoding order is:

$I_1\ P_4\ B_2\ B_3\ P_6\ B_5\ P_8\ B_7\ P_{11}\ B_9\ B_{10} \ldots P_{n-1}\ B_{n-2}\ I_n.$

The decoding order differs from the display order because the B-frames require future I- or P-frames for their decoding. Fig. 2 displays the beginning of the above frame sequence and can be referred to in order to understand the dependencies of the frames, as described earlier. P-frames require the previous I- or P-reference frame be

available. For example, P_4 requires I_1 to be decoded. Similarly, frame P_6 requires that P_4 be available in order to decode/decompress frame P_6. B-frames, such as frame B_3, require a past and/or a future I- or P-reference frame, such as P_4 and I_1 in order to be decoded. B-frames are frames between I-or P-frames during encoding.

A prior art system for encoding and decoding is shown in Figures 3 and 4. Referring to the encoder 300 of Figure 3, the frame 301 being coded I(x,y), called the current frame, is partitioned into rectangular regions of KxL pixels. The coordinates (x,y) denote the location of the pixels within the frame. Each block is encoded using either intra coding (i.e. using only the spatial correlation of the image data within the block) or inter coding (i.e. utilising both spatial and temporal prediction). The following description considers the process by which inter-coded blocks are formed. Each inter-coded block is predicted 360 from one of the previously (or subsequently) coded and transmitted frames R(x,y) in the frame memory 350, called a reference frame. The motion information used for the prediction is obtained from motion estimation and coding block 370 by using the reference frame and the current frame 305. The motion information is represented by a two-dimensional motion vector (Δx, Δy) where Δx is a horizontal and Δy is a vertical displacement. In the motion compensated (MC) prediction block, the motion vectors are used together with the reference frame to construct prediction frame P(x,y):

$$P(x,y) = R(x+\Delta x, y+\Delta y).$$

Subsequently the prediction error E(x,y), i.e. the difference between the current frame and the prediction frame P(x,y), is calculated 307 according to

$$E(x,y)= I(x,y)-P(x,y).$$

In transform block 310, the prediction error for each KxL block is represented as weighted sum of a transform basis functions $f_{ij}(x,y)$,

$$E(x, y) = \sum_{i=1}^{K} \sum_{j=1}^{L} c_{err}(i,j)\, f_{ij}(x, y).$$

The weights $c_{err}(i,j)$ corresponding to the basis functions are called transform coefficients. These coefficients are subsequently quantised in quantisation block 320 to give

$$I_{err}(i,j)=Q(c_{err}(i,j),QP)$$

where $I_{err}(i,j)$ are the quantised transform coefficients. The quantisation operation Q() introduces loss of information, but the quantised coefficients can be represented with smaller number of bits. The level of compression (loss of information) is controlled by adjusting the value of the quantisation parameter QP.

Before the motion vectors and quantised transform coefficients are supplied to the multiplexer 380, they are further encoded using Variable Length Codes (VLC). This reduces the number of bits needed to represent the motion vectors and quantised transform coefficients. The encoded motion vectors, quantised transform coefficients as well as other additional information needed to represent each coded frame is multiplexed in the multiplexer 380, and the resulting bit-stream is transmitted 415 to the decoder. The quantised transform coefficients are also forwarded to the inverse quantisation block 330 to obtain inverse quantised transform coefficients and further to the inverse transform block 340 to obtain the prediction error information $E_c(x,y)$ for the current frame. The prediction error information $E_c(x,y)$ is added to the predicted frame P(x,y) in a summing element to obtain a video frame that can be subsequently stored in the frame memory 350.

In the following, decoding video frames will be described with reference to Figure 4. The decoder 400 receives the multiplexed video bit-stream 415 from the encoder, and the demultiplexer 410 demultiplexes the bit-stream to obtain the constituent parts of the video frames to be decoded. These parts comprise at least the coded quantised prediction

error transform coefficients and the coded motion vectors, which are subsequently decoded (not shown) to obtain quantised prediction error transform coefficients and motion vectors. The quantised transform coefficients are inverse quantised in inverse quantisation block 420 to obtain inverse quantised transform coefficients $d_{err}(i,j)$ according to the following relation:

$$d_{err}(i,j)=Q^{-1}(I_{err}(i,j), QP).$$

In inverse transform block 430, the inverse quantised transform coefficients are subjected to an inverse transform to obtain prediction error $E_c(x,y)$:

$$E_c(x,y) = \sum_{i=1}^{K}\sum_{j=1}^{L} c_{err}(i,j)\, f_{ij}\,(x,y).$$

The pixels of the current coded frame are reconstructed by finding the prediction pixels in the reference frame R(x,y) obtained from frame memory 440, and using the received motion vectors together with the reference frame in motion compensation prediction block 450, to obtain a predicted frame P(x,y). The predicted frame P(x,y) and the prediction error information $E_c(x,y)$ are summed together in a summing element 435 according to the following relation:

$$I_c(x,y)= R(x+\Delta x, y+\Delta y)+E_c(x,y).$$

These values $I_c(x,y)$ can further be filtered to obtain decoded video frames 445. The values $I_c(x,y)$ are also stored in the frame buffer 440. The reconstructed values $I_c(x,y)$ can be filtered in a filtering block (not shown in Figure 4) after the summation block 435.

Video streaming has emerged as an important application in the fixed Internet. It is further anticipated that video streaming will also be important in and in the near future in 3G wireless networks. In streaming applications the transmitting server starts transmitting a pre-

encoded video bit stream via a transmission network to a receiver upon a request from the receiver. The receiver plays back the video stream back while receiving it. The best-effort nature of present networks causes variations in the effective bandwidth available to a user due to the changing network conditions. To accommodate these variations, the transmitting server can scale the bit rate of the compressed video. In the case of a conversational service characterized by real-time encoding and point-to-point delivery, this can be achieved by adjusting the source encoding parameters on the fly. Such adjustable parameters can be, for example, a quantisation parameter, or a frame rate. The adjustment is advantageously based on feedback from the transmission network. In typical streaming scenarios when a previously encoded video bit stream is to be transmitted to the receiver, the above solution cannot be applied.

One solution to achieve bandwidth scalability in case of pre-encoded sequences is to produce multiple and independent streams having different bit-rates and quality. The transmitting server then dynamically switches between the streams to accommodate variations in the available bandwidth. The following example illustrates this principle. Let us assume that multiple bit streams are generated independently with different encoding parameters, such as quantisation parameter, corresponding to the same video sequence. Let $\{P_{1,n-1}, P_{1,n}, P_{1,n+1}\}$ and $\{P_{2,n-1}, P_{2,n}, P_{2,n+1}\}$ denote the sequence of decoded frames from bit streams 1 and 2, respectively. Since the encoding parameters are different for the two bit streams, frames reconstructed from them at the same time instant, for example, frames $P_{1,n-1}$ and $P_{2,n-1}$, are not identical. If it is now assumed that the server initially sends encoded frames from bit stream 1 up to time *n* after which it starts sending encoded frames from bit stream 2, the decoder receives frames $\{P_{1,n-2}, P_{1,n-1}, P_{2,n}, P_{2,n+1}, P_{2,n+2}\}$. In this case $P_{2,n}$ cannot be correctly decoded since its reference frame $P_{2,n-1}$ is not received. On the other hand, the frame $P_{1,n-1}$, which is received instead of $P_{2,n-1}$, is not identical to $P_{2,n-1}$. Therefore switching between bit streams at arbitrary locations leads to visual artefacts due to the mismatch between the reference frames used for motion compensated prediction in the different sequences.

These visual artefacts are not only confined to the frame at the switching point between bit streams, but propagates in time due to the continued motion compensated coding in the remaining part of the video sequence.

In current video encoding standards, perfect (mismatch-free) switching between bit streams is only possible at positions where the current and future frames or regions thereof do not use any information previous to the current switching location, i.e., at I-frames. Furthermore, by placing I-frames at fixed (e.g. 1 sec) intervals, VCR functionalities, such as random access or "Fast Forward" and "Fast Backward" (increased playback rate) for streaming video content, are achieved. The user may skip a portion of the video sequence and restart playing at any I-frame location. Similarly, an increased playback rate can be achieved by transmitting only I–frames. The drawback of using I-frames in these applications is that since I-frames do not exploit any temporal redundancy they require a much larger number of bits than P-frames at the same quality.

SUMMARY OF THE INVENTION

It is an aim of the present invention to provide a new method and a system for transmitting video images in variable transmission environments. The invention is based on the idea that correct (mismatch-free) switching between video streams is enabled by forming a new type of a compressed video frame and inserting frames of the new type into video bit-streams at locations where switching from one bit-stream to another is to be allowed. In this description, the new type of compressed video frame will be referred to generally as an S-frame. More specifically, S-frames may be classified as SP-frames, which are formed at the decoder using motion compensated prediction from already decoded frames using motion vector information, and SI-frames, which are formed at the decoder using spatial (intra) prediction from already decoded neighbouring pixels within a frame being decoded. In general, an S-frame according to the invention is formed on a block-by-block basis and may comprise both inter-coded (SP)

blocks as well as intra-coded (SI) blocks. The method according to the present invention is primarily characterized in that in switching the transmission over from said first bit-stream to said second bit-stream, the secondary bit-stream comprises at least one primary switching frame, a secondary switching frame is transmitted, the secondary switching frame having been encoded using the encoding parameters of the second bit-stream and at least one reference frame from the first bit-stream, and said secondary switching frame is used in place of the primary switching frame as a reference frame to be used in the reconstruction of said at least one predictive video frame of the second set of video frames. The encoder according to the present invention is primarily characterized in that said means for switching the transmission over from said first bit-stream to said second bit-stream comprise means for encoding a secondary switching frame using the encoding parameters of the second bit-stream and reference frames from the first bit-stream to enable to switch the transmission over from said first bit-stream to said second bit-stream. The decoder according to the present invention is primarily characterized in that the decoder comprises means for decoding a secondary switching frame, said secondary switching frame having been encoded using the encoding parameters of the second bit-stream and at least one reference frame from the first bit-stream, and added to said signal in place of a primary switching frame as a reference frame to be used in the reconstruction of said at least one predictive video frame of the second set of video frames, and that said means for decoding a secondary switching frame comprise means for using decoding parameters of the second bit-stream and reference frames from the first bit-stream. The signal according to the present invention is primarily characterized in that in switching the transmission over from said first bit-stream to said second bit-stream, the secondary bit-stream comprises at least one primary switching frame, said signal comprises a secondary switching frame having been encoded using the encoding parameters of the second bit-stream and at least one reference frame from the first bit-stream, and said secondary switching frame is in place of the primary switching frame as a reference frame to be used in the reconstruction

of said at least one predictive video frame of the second set of video frames.

Considerable advantages are achieved by the present invention compared to the methods and systems of prior art. The invention allows switching between bit streams to occur not only at the locations of I-frames but also at the locations of the SP-frames. The coding efficiency of an SP-frame is much better than the coding efficiency of a typical I-frame wherein less bandwidth is needed to transmit bit streams having SP-frames in locations where, according to prior art, I-frames would be used, while still providing enough adaptability to changing transmission conditions. The switching of one bit stream into another can be performed at locations in which an SP-frame according to the invention is placed in the encoded bit stream. The images reconstructed from the bit stream by the decoder is not degraded as a result of changing from one bit stream to another. The invention also has the advantage that random access, fast-forward and fast rewind operations can be performed on the bit stream. The system according to the invention offers improved error recovery and resiliency properties compared with the described prior art solutions.

These and other features, aspects, and advantages of embodiments of the present invention will become apparent with reference to the following description in conjunction with the accompanying drawings. It is to be understood, however, that the drawings are designed solely for the purposes of illustration and not as a definition of the limits of the invention, for which reference should be made to the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

In the following the invention will be described in more detail with reference to the appended drawings, in which

Figs. 1A-1C and 2 are diagrams showing the prior art encoding/compression of video frames,

Fig. 3 is a block diagram of a generic motion-compensated predictive video coding system (encoder),

Fig. 4 is a block diagram of a generic motion-compensated predictive video coding system (decoder),

Fig. 5 is an illustration showing switching between two different bit streams using S-frames according to the invention,

Fig. 6 is a block diagram of a decoder in accordance with an advantageous embodiment of the invention,

Fig. 7 is an illustration of random access using S-frames,

Fig. 8 is an illustration of a fast-forward process using S-frames,

Fig. 9 is a block diagram of a decoder in accordance with another advantageous embodiment of the invention,

Fig. 10 is a block diagram of an decoder in accordance with yet another advantageous embodiment of the invention,

Fig. 11 is a block diagram of an encoder in accordance with an advantageous embodiment of the invention,

Fig. 12 is a block diagram of a decoder in accordance with another advantageous embodiment of the invention,

Fig. 13 is a block diagram of an encoder according to a second embodiment of the invention,

Fig. 14 is an illustration of a error resiliency/recovery process using SP-frames,

Fig. 15 is a block diagram of an encoder in accordance with the third advantageous embodiment of the invention,

Fig. 16 is a block diagram of a decoder in accordance with yet another advantageous embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following the invention is described in a system in which multiple bit streams are formed from a video signal. The video signal can be any digital video signal comprising multiple images, *i.e.* an image sequence. The digital video signal is encoded in an encoder to form multiple bit streams. Each bit stream is encoded from the same video signal using at least partly different encoding parameters. For example, the bit rate can be altered by selecting the encoding parameters differently, and in this way bit streams with different bit rates can be formed. The encoding parameters can be, for example, frame rate, quantisation parameter, spatial resolution, or another factor affecting the images size, known as such to a person skilled in the art. The encoder also inserts at least one Intra frame to each bit stream. Typically, at least the first frame of each bit stream is preferably an Intra frame. This enables the decoder to start reconstruction of the video signal. The encoder used in the encoding of the I-frames, P-frames and B-frames can be any prior art encoder which performs the encoding of the video signal, or there may exist more than one prior art encoder each using different encoding parameters to form multiple bit streams. However, to encode a video signal containing also SP-frames and/or SI-frames according to the invention, new functionality is needed in the encoder. This will be explained later in greater detail.

The encoder also inserts frames encoded using motion compensated predictive coding (P-frames and optionally B-frames) into the bit streams. The encoder also inserts a new type of frames, referred to as S-frames in this description, into each bit stream at locations where switching between different bit streams will be allowed according to the invention. The S-frames may be used at locations where in prior art methods an Intra coded frame would be inserted, or the S-frames may be used in addition to using Intra coded frames in the video sequence.

Advantageously, the different bit streams are stored into storage means for later use. However, it is also possible that transmission may take place substantially immediately after encoding wherein it is not necessary to store complete video sequences, but storing the necessary reference frames suffices. Transmission of the encoded video stream may be performed *e.g.* by a transmitting server, which has means for retrieving the stored bit streams for transmission, and/or means for receiving the bit streams directly from the encoder. The transmitting server also has means for transmitting the bit stream to the transmission network (not shown).

In the following, a method according to an advantageous embodiment of the invention will be described. Fig. 5 depicts part of a first bit stream 510 and part of a second bit stream 520, which are formed in the encoder. Only a few P-frames of the respective bit streams are shown. Specifically, the first bit stream 510 is shown to comprise P-frames 511, 512, 514 and 515, while the second bit stream 520 comprises corresponding P-frames 521, 522, 524 and 525. Both the first bit stream 510 and the second bit stream 520 also comprise S-frames 513 (also labelled S_1), 523 (also labelled S_2) at corresponding locations. It is assumed that the two bit streams 510 and 520 correspond to the same sequence encoded at different bit rates for example by using different frame rates, different spatial resolutions or different quantisation parameters. It is further assumed that the first bit stream 510 is being transmitted from the transmitting server to a decoder 600, 1200, 1300 (Figs. 6, 12 and 13, respectively) via a transmission network, and that the transmitting server receives a request from the transmission network to change the bit rate of the video stream being transmitted.

As mentioned above, S-frames are placed in the bit stream during the encoding process at those locations within the video sequences where switching from one bit stream to another is allowed. As can be seen from Figure 5, in an advantageous embodiment of the invention a further S-frame 550 (also labelled S_{12}) is associated with S-frames S_1 and S_2. This S-frame is called a secondary representation of the S-

frame (or secondary S-frame for short), and is only transmitted during bit stream switching. This secondary S-frame S_{12} is generated by a special encoding of the nth frame in the video sequence using the encoding parameters of the corresponding S-frame 523 (S_2) in the second bit stream 520 and using motion compensated prediction from the reference frames of the nth frame in the first bit stream 510. It should be noted that in the situation shown in Figure 5, S-frame S_2 uses the previously reconstructed frames from the second bit stream 520 as reference frames and the secondary S-frame S_{12} uses the previously reconstructed frames from the first bit stream 510 as reference frames. However, the reconstructed pixel values of both S_2 and S_{12} are identical. S-frame S_{12} is transmitted only when switching from the first bit stream 510 to the second bit stream 520 is actually performed. Therefore it is not necessary to form the secondary S-frames during the encoding phase but only when switching is performed. On the other hand, it may be useful to have at least some secondary S-frames formed earlier at the time when the different bit streams are formed to reduce computational load during transmission.

When the transmitting server reaches the frame of the video sequence encoded as S-frame 513 (S_1) in the first bit stream 510, it can begin the necessary operations to continue transmission of the video stream using the encoded frames of the second bit stream 520. At that point the transmitting server has already transmitted P-frames 511 and 512 from the first bit stream 510 and the decoder 600, 1200, 1300 has received and decoded the respective P-frames 511, 512. Thus, those frames have already been stored in the frame memory 640, 1250, 1360 of the decoder 600, 1200, 1300. The frame memory 640, 1250, 1360 comprises sufficient memory to store all those frames, which are needed to reconstruct a P-frame or a B-frame, *i.e.* the necessary information of all the reference frames required by the current frame to be reconstructed.

The transmitting server performs the following operations to continue the transmission of the video stream using the encoded frames of the second bit stream 520. The transmitting server notices, for example, by

examining the type information of the frame, that the current frame to be transmitted is an S-frame, so it is possible to perform switching between the bit streams. Of course, switching is only performed if a request to do so has been received or there is for some other reason a need to perform the switch. The transmitting server inputs the corresponding S-frame 523 of the second bit stream, uses it to form the secondary S-frame 550 (S_{12}), and transmits the secondary S-frame (S_{12}) to the decoder 600, 1200, 1300. The transmitting server does not transmit the S-frame S_2 of the second bit stream, but sends the secondary S-frame S_{12} in its place. The secondary S-frame is formed in such a way that by decoding the secondary S-frame S_{12} the decoder 600 can reconstruct an image identical to that which would be produced if it were using respective frames 521, 522 and the S-frame 523 of the second bit stream 520. After the transmission of the secondary S-frame the transmitting server continues to transmit the encoded frames if the second bit stream 520, i.e., 524, 525, and so on.

The S-frame 513, 523, 550 can comprise blocks encoded using only spatial correlation among the pixels (intra blocks) and blocks encoded using both spatial and temporal correlation (inter blocks). For each inter block the prediction of this block, P(x,y), is formed in the decoder 600, 1200, 1300 using received motion vectors and a reference frame. The transform coefficients c_{pred} for P(x,y) corresponding to basis functions $f_{ij}(x,y)$ are calculated and quantised. The quantised values of the transform coefficients c_{pred} are denoted as l_{pred} and the dequantised values of the quantised transform coefficients l_{pred} are denoted as d_{pred}. Quantised coefficients l_{err} for the prediction error are received from the encoder. The dequantised values of these coefficients will be denoted as d_{err}. Value of each pixel S(x,y) in the inter block is decoded as a weighted sum of the basis functions $f_{ij}(x,y)$ where the weight values d_{rec} will be called dequantised reconstruction image coefficients. The values of d_{rec} have to be such that coefficients c_{rec} exist from which by quantisation and dequantisation d_{rec} can be obtained. In addition, values d_{rec} have to fulfill one of the following conditions:

$d_{rec} = d_{pred} + d_{err}$, or

$c_{rec} = c_{pred} + d_{err}$.

Values S(x,y) can further be normalized and filtered.

Next, encoding of the S-frames placed within the bit stream, e.g. S-frames 513 (S_1) and 523 (S_2) is described.

Generally, an S-frame according to the invention such as frames 513, 523 in Figure 5 is constructed on a block-by-block basis. As explained earlier, each of the blocks may be coded either in such a way as to take advantage of the spatial correlations among pixels of the image being coded (intra or SI-blocks) or in such a way as to take advantage of the temporal correlation between blocks of pixels in successive frames of a video sequence (inter or SP-blocks).

The encoding of S-frames according to the invention will be described with reference to Figure 11 which is a block diagram of an S-frame encoder 1100 according to a first embodiment of the invention.

A video frame to be encoded in S-frame format is first partitioned into blocks and each block is then encoded as either an SP-block, an SI-block, or an intra-block, an intra-block being known as such from prior art. Switch 1190 is operated as appropriate to switch between the SI and SP encoding modes, i.e., the switch 1190 is a construction used in the description of the invention, not necessarily a physical device. In SP-encoding mode switch 1190 is operated to obtain a motion compensated prediction for the current block 1170. Motion compensated prediction block 1170 forms a prediction P(x,y) for the current block of the frame being encoded in a manner analogous to that used in motion compensated prediction known from prior art. More specifically, motion compensated prediction block 1170 forms the prediction P(x,y) for the current block of the frame being encoded by determining a motion vector describing the relationship between the pixels in the current block and pixel values of a reconstructed reference frame held in frame memory 1146.

In SI-encoding mode switch 1190 is operated to obtain a prediction for the current block of the frame being coded from intra prediction block 1180. Intra prediction block 1180 forms the prediction P(x,y) for the current block of the frame being encoded in a manner analogous to that used in intra prediction known from prior art. More specifically, intra prediction block 1180 forms the prediction P(x,y) for the current block of the frame being encoded using spatial prediction from already encoded neighbouring pixels within the frame being encoded.

In both SP- and SI-coding modes the prediction P(x,y) takes the form of a block of pixel values. A forward transform, for example a Discrete Cosine Transform (DCT), is applied to the predicted block of pixel values P(x,y) in block 1160 and the resulting transform coefficients, referred to as c_{pred}, are subsequently quantised in quantisation block 1150 to form quantised transform coefficients l_{pred}. Corresponding operations are also performed on the original image data. More specifically, the current block of pixel values of the original image being encoded is applied to transform block 1110. Here, a forward transform (e.g., a DCT) is applied to the pixel values of the original image block to form transform coefficients c_{orig}. These transform coefficients are passed to quantisation block 1120 where they are quantised to form quantised transform coefficients l_{orig}. The summing element 1130 receives both sets of quantised transform coefficients l_{pred} and l_{orig} from the respective quantisation blocks 1150 and 1120 and generates a set of quantised prediction error coefficients l_{err} according to the relationship:

$$l_{err} = l_{orig} - l_{pred} .$$

The quantised prediction error coefficients l_{err} are passed to multiplexer 1135. If the current block is encoded in SP-format/mode, multiplexer 1135 also receives the motion vectors for the SP-coded block. If the current block is encoded in SI-format/mode, information concerning the intra prediction mode used to form the prediction for the SI-coded block in intra prediction block 1180 is passed to the multiplexer. Advantageously, variable length coding is applied to the quantised

prediction error coefficients I_{err} and to the motion vector or intra prediction mode information in the multiplexer 1135, a bit-stream is formed by multiplexing together the various forms of information and the bit-stream thus formed is transmitted to a corresponding decoder 1200, 1300 (see Figs. 12 and 13).

The S-frame encoder 1100 according to the invention also comprises local decoding functionality. The quantised prediction transform coefficients I_{pred} formed in quantisation block 1150 are supplied to the summing element 1140 which also receives the quantisation error coefficients I_{err} . The summing element 1140 recombines the quantised prediction transform coefficients I_{pred} and the quantised prediction error coefficients I_{err} to form a set of reconstructed quantised transform coefficients I_{rec} according to the relationship:

$$I_{rec} = I_{pred} + I_{err} .$$

The reconstructed quantised transform coefficients are passed to inverse quantisation block 1142 which inverse quantises the reconstructed quantised transform coefficients to form inverse quantised reconstructed transform coefficients d_{rec}. The inverse quantised reconstructed transform coefficients are further passed to inverse transform block 1144 where they are subjected to e.g. an Inverse Discrete Cosine Transform (IDCT), or any other inverse transform corresponding to the transform performed in block 1160. As a result, a block of reconstructed pixel values is formed for the image block in question and is stored in frame memory 1146. As subsequent blocks of the frame being encoded in S-frame format undergo the previously described encoding and local decoding operations, a decoded version of the current frame is progressively assembled in the frame memory from where it can be accessed and used in intra prediction of subsequent blocks of the same frame or in inter (motion compensated) prediction of subsequent frames in the video sequence.

Operation of a generic S-frame decoder according to a first embodiment of the invention will now be described with reference to Figure 12.

The bit-stream generated by the S-frame encoder previously described in connection with Figure 11 is received by decoder 1200 and is demultiplexed into its constituent parts by demultiplexer 1210. The decoder reconstructs a decoded version of the S-frame on a block-by-block basis. As previously described, an S-frame may comprise intra-blocks, SP-coded and SI-coded image blocks. For SP-format image blocks, the information in the received bit-stream comprises VLC encoded motion coefficient information and VLC encoded quantised prediction error coefficients I_{err}. For image blocks encoded in SI-format the information in the received bit-stream comprises VLC coded information relating to the intra prediction mode used to form the intra prediction for the SI-coded block together with VLC coded quantised prediction error coefficients I_{err}.

When decoding an SP-coded block, demultiplexer 1210 first applies appropriate variable length decoding (VLD) to the received bit-stream to recover the motion vector information and quantised prediction error coefficients I_{err}. It then separates the motion vector information from the quantised prediction error coefficients I_{err}. The motion vector information is supplied to motion compensated prediction block 1260 and the quantised prediction error coefficients recovered from the bit-stream are applied to one input of summing element 1220. The motion vector information is used in motion compensated prediction block 1260 together with pixel values of a previously reconstructed frame held in frame memory 1250 to form a prediction P(x,y) in a manner analogous to that employed in the encoder 1100.

When decoding an SI-coded block, demultiplexer 1210 applies appropriate variable length decoding to the received intra prediction mode information and the quantised prediction error coefficients I_{err}. The intra prediction mode information is then separated from the quantised prediction error coefficients and supplied to intra prediction

block 1270. The quantised prediction error coefficients I_{err} are supplied to one input of the summing element 1220. The intra prediction mode information is used in intra prediction block 1270 in conjunction with previously decoded pixel values of the current frame held in frame memory 1250 to form a prediction P(x,y) for the current block being decoded. Again, the intra prediction process performed in decoder 1200 is analogous to that performed in encoder 1100 and previously described.

Once a prediction for the current block of the frame being decoded has been formed, switch 1280 is operated so that the prediction P(x,y) which comprises predicted pixel values is supplied to transform block 1290. Again, switch 1280 is an abstract construction used in the description of the invention, not necessarily a physical device. In the case of an SP-coded block, switch 1280 is operated to connect motion compensated prediction block 1260 to transform block 1290, while in the case of an SI-coded block it is operated to connect intra prediction block 1270 to transform block 1290.

In block 1290, a forward transform, e.g., a Discrete Cosine Transform (DCT), is applied to the predicted block of pixel values P(x,y) and the resulting transform coefficients c_{pred} are supplied to quantisation block 1295 where they are quantised to form quantised transform coefficients I_{pred}. Quantised transform coefficients I_{pred} are then supplied to the second input of summing element 1220 where they are added to the prediction error coefficients I_{err} to form reconstructed quantised transform coefficients I_{rec} according to the relationship:

$$I_{rec} = I_{pred} + I_{err}.$$

The reconstructed quantised transform coefficients I_{rec} are further supplied to inverse quantisation block 1230 where they are inverse quantised to form inverse quantised reconstructed transform coefficients d_{rec}. The inverse quantised transform coefficients d_{rec} are then passed to inverse transform block 1240 where they are subjected to e.g. an Inverse Discrete Cosine Transform (IDCT), or any other

inverse transform corresponding to the transform performed in block 1290. In this way, a block of reconstructed pixel values is formed for the image block in question. The reconstructed pixel values are supplied to the video output and to frame memory 1250. As subsequent blocks of the S-frame being decoded undergo the previously described decoding operations, a decoded version of the current frame is progressively assembled in frame memory 1250 from where it can be accessed and used in the intra prediction of subsequent blocks of the same frame or in the inter (motion compensated) prediction of subsequent frames in the video sequence.

Having reviewed the structure and function of an S-frame encoder and decoder according to a first embodiment of the invention, it is now possible to understand how S-frames according to the invention make it possible to switch between bit-streams without mismatch errors such as those encountered in prior video encoding/decoding systems. Referring once more to the bit-stream switching example depicted in Figure 5, switching from the first bit-stream 510 to the second bit-stream 520 occurs at the location of S-frames S_1 (513) and S_2 (523) in the respective bit-streams. As previously noted, when switching is performed, a secondary S-frame, denoted S_{12} (550) is encoded and transmitted. The secondary S-frame S_{12} is encoded using the encoding parameters of the second bit-stream 520 and reference frames from the first bit-stream 510 in such a way that when the secondary frame S_{12} is decoded its reconstructed pixel values are identical to those that would have resulted from transmission of frame S_2 in the second bit-stream.

Let I^2_{err} and I^2_{pred} denote the quantised coefficients of the prediction error and the prediction frame, respectively, obtained from encoding of the SP-frame S_2 with the procedure described above, and let I^2_{rec} denote the quantised reconstructed image coefficients of the S-frame S_2. The encoding of the secondary S-frame 550 (S_{12}) follows the same procedures as in the encoding of the S-frame 523 (S_2) with the following exceptions: 1) The reference frame(s) used in the prediction of each block of the secondary S-frame S_{12} are the reconstructed

frames obtained by decoding the first bit stream 510 up to the current n^{th} frame in the video sequence, 2) The quantised prediction error coefficients are calculated as follows: $I^{12}_{err} = I^{2}_{rec} - I^{12}_{pred}$ where I^{12}_{pred} denotes the quantised prediction transform coefficients. The quantised prediction error coefficients I^{12}_{err} and the motion vectors are transmitted to the decoder 1200.

When decoding the secondary S-frame S_{12} in the decoder 1200 using the reconstructed frames from the first bit stream 510 before the switch as reference frames, coefficients I^{12}_{pred} of the secondary S-frame are constructed and added to the received quantised prediction error coefficients I^{12}_{err} as described above, i.e., $I^{12}_{rec} = I^{12}_{err} + I^{12}_{pred} = I^{2}_{rec} - I^{12}_{pred} + I^{12}_{pred} = I^{2}_{rec}$. It can be noted from this equation that I^{12}_{rec} and I^{2}_{rec} are identical. Therefore, although the secondary S-frame S_{12} and the S-frame of the second bit stream S_2 have different reference frames, when S_{12} is decoded it produces an image with reconstructed pixel values identical to those that would result from decoding S-frame S_2.

From the foregoing description of the encoding and decoding of S-frames according to the invention, it will be appreciated that a significant difference exists compared with encoding and decoding of P-frames and I-frames according to prior art, specifically, it should be appreciated that when encoding or decoding an image block in SP or SI -format, the prediction P(x,y) for the block in question is transformed into the transform coefficient domain by applying a transformation such as a Discrete Cosine Transform. The transform coefficients thus produced are subsequently quantised and a prediction error is determined in the quantised coefficient domain. This contrasts with predictive coding according to prior art in which a prediction error is determined in the spatial (pixel value) domain.

In the following, operation of the decoder 1200 during switching between bit-streams 510 and 520 is described in detail. At the position in the video sequence where switching from the first bit-stream to the second bit-stream 520 takes place, the decoder 1200 has already

received and decoded the previous P-frames 511 and 512 of the first bit-stream 510. The decoded frames are stored in frame memory 1250 and are thus available for use as reference frames. When switching from the first bit-stream 510 to the second bit-stream 520 takes place, encoder 1100 (figure 11) constructs and encodes secondary S-frame S_{12} (550) and transmits the encoded video information representing S_{12} to decoder 1200.

As previously explained, encoding is performed on a block-by-block basis. Specifically, secondary S frame S_{12} is encoded as an assembly of image blocks and generally each image block is encoded as either an SP-coded block or an SI-coded block or an intra-block. For the SP-coded blocks of the secondary S-frame S_{12}, the compressed video information transmitted from encoder to decoder takes the form of quantised prediction error transform coefficients I^{12}_{err} and motion vector information. For the SI-coded blocks of the secondary S-frame S_{12} the compressed video information comprises quantised prediction error transform coefficients I^{12}_{err} and information relating to the intra prediction mode used in forming the prediction for the SI-coded block in the encoder. As previously explained, the compressed video information is subjected to appropriate variable length coding (VLC) prior to transmission from the encoder in order to further reduce the number of bits required for its representation.

The compressed video information for a given image block is received at decoder 1200 and is first subjected to appropriate variable length decoding (VLD) and is separated into its constituent parts by demultiplexer 1210. The quantised prediction error coefficients I^{12}_{err} extracted from the received bit-stream are applied to the first input of summer 1220 and a block of predicted pixel values P(x,y) is formed for each image block according to its coding mode (SP or SI). In the case of an SP-coded block, the block of predicted pixel values P(x,y) is formed in motion compensated prediction block 1260 using a reference frame from the first bit-stream (e.g. P-frame 511 or 512) available in frame memory 1250 and the motion vector information extracted from the encoded video information of secondary S-frame S_{12} by

demultiplexer 1210. In the case of an SI-coded block, a block of predicted pixel values P(x,y) is formed in intra prediction block 1270 using previously decoded pixels of the secondary S-frame S_{12} also stored in frame memory 1250. Intra prediction is performed according to the intra prediction mode information extracted from the received video information for secondary S-frame S_{12} by demultiplexer 1210.

Once a prediction for the current block of the secondary S-frame has been formed the predicted pixel values P(x,y) are passed to transform block 1290. Here a forward transform (e.g. a Discrete Cosine Transform (DCT)) is applied the predicted pixel values P(x,y) to form a set of transform coefficients c_{pred}. These transform coefficients are then passed to quantisation block 1295 where they are quantised to form quantised transform coefficients l^{12}_{pred}. The quantised transform coefficients l^{12}_{pred} are, in turn, applied to the second input of summer 1220. Summer 1220 combines the quantised transform coefficients l^{12}_{pred} with quantised prediction error transform coefficients l^{12}_{err} to form reconstructed quantised transform coefficients l^{12}_{rec} according to the relationship:

$$l^{12}_{rec} = l^{12}_{pred} + l^{12}_{err}.$$

The reconstructed quantised transform coefficients l^{12}_{rec} are then supplied to inverse quantisation block 1230 where they are inverse quantised to form inverse quantised reconstructed transform coefficients d^{12}_{rec}. The inverse quantised reconstructed transform coefficients d^{12}_{rec} are then passed to inverse transform block 1240 where they are subjected to an inverse transform operation (e.g. an Inverse Discrete Cosine Transform (IDCT)). As a result, a block of reconstructed pixel values is formed for the current block of the secondary S-frame S_{12}. The reconstructed pixel values $I_c(x,y)$ are supplied to the video output and to frame memory 1250. As subsequent blocks of the secondary S-frame S_{12} are encoded, transmitted from encoder 1100 to decoder 1200 and subsequently decoded, a decoded version of the secondary S-frame is progressively built up in frame memory 1250. From there, the already decoded

blocks of the secondary S-frame can be retrieved and used by intra prediction block 1270 to form predicted pixel values P(x,y) for subsequent blocks of the secondary S-frame S_{12}. Here it should be remembered that the quantised prediction error transform coefficients for each image block of the secondary S-frame S_{12} are generated in encoder 1100 according to the relationship:

$$I^{12}_{err} = I^{2}_{rec} - I^{12}_{pred}$$

where I^{2}_{rec} are quantised reconstructed transform coefficient values produced by encoding and subsequently decoding S-frame S_2 in the second bit-stream. This means that the reconstructed transform coefficients I^{12}_{rec} produced by decoding the compressed video information for secondary S-frame S_{12} are identical to those which would have been produced if S-frame S_2 from the second bit-stream had been transmitted and decoded. As previously explained, this follows since:

$$I^{12}_{rec} = I^{12}_{pred} + I^{12}_{err}$$

$$= I^{12}_{pred} + I^{2}_{rec} - I^{12}_{pred} = I^{2}_{rec}$$

Thus, $I^{12}_{rec} = I^{2}_{rec}$.

Therefore it can be seen that by constructing a secondary S-frame S_{12} according to the method of the invention, transmitting it from encoder to decoder and subsequently decoding it, mis-match-free switching between a first and a second bit-stream can be achieved.

Consider the case that the secondary S-frame is an SI-frame while the S-frame within the bit stream is an SP-frame. In this case, a frame which uses motion-compensated prediction is represented by a frame which uses only spatial prediction. This special case has implications in random access and error resiliency which will described later.

It should be noted that in encoder 1100 and decoder 1200 according to the first embodiment of the invention, described above, the quantisation applied to the transform coefficients c_{pred} produced in transform blocks 1160 (encoder) and 1290 (decoder) to yield quantised transform coefficients l_{pred} is the same as that used to generate the quantised prediction error transform coefficients l_{err}. More specifically, in the first embodiment of the invention, when a block of predicted pixel values P(x,y) is produced for an image block of an S-frame which is being encoded/decoded, the quantisation parameter QP used to quantise the transform coefficients c_{pred} corresponding to the predicted block of pixel values P(x,y) should be the same as the quantisation parameters used in generating the quantised prediction error transform coefficients l_{err}. This is advantageous since the summation performed in order to produce reconstructed transform coefficients l_{rec} is performed in the quantised transform coefficient domain, i.e. since

$$l_{rec} = l_{pred} + l_{err},$$

failure to use identical quantisation parameters in the construction of l_{pred} and l_{err} will lead to errors in the reconstructed quantised transform coefficients l_{rec}.

Figure 15 shows a block diagram of an S-frame encoder 1500 according to a second embodiment of the invention which provides greater flexibility in the choice of quantisation parameters to generate the quantised transform coefficients l_{pred} and l_{err}. As can be seen by comparing Figure 15 with Figure 11, the principal difference between the S-frame encoder 1500 according to a second embodiment of the invention and S-frame encoder 1100 according to a first embodiment of the invention concerns the location of quantisation blocks 1525 and 1550. The operation of the S-frame encoder 1500 according to the second embodiment of the invention will now be described in detail with reference to Figure 15.

According to the second embodiment of the invention a video frame to be encoded in S-frame format is first partitioned into blocks and each

block is then encoded as either on SP-block or on SI-block. Switch 1585 is operated as appropriate to switch between the SP and SI coding modes. In SP coding mode switch 1585 is operated to obtain a motion compensated prediction for the current block of the frame being encoded from motion compensated prediction block 1575. Motion compensated prediction block 1575 forms a block of predicted pixel values P(x,y) for the current block of the frame being encoded by determining a motion vector describing the relationship between the pixels of the current block and pixel values of a reconstructed reference frame held in frame memory 1570.

In SI-coding, mode switch 1585 is operated to obtain a prediction for the current block of the frame being encoded from intra prediction block 1580. Intra prediction block 1580 operates in a manner analogous to that described in connection with the first embodiment of the invention to form a block of predicted pixel values P(x,y) for the current block of the frame being encoded using spatial prediction from already encoded neighbouring pixels within the frame being encoded.

In both SP- and SI-coding modes a forward transform, for example a Discrete Cosine Transform (DCT), is applied to the predicted block of pixel values P(x,y) in transform block 1590. The resulting transform coefficients c_{pred} are supplied to summers 1520 and 1540. The original image data, comprising the actual pixel values of the image block currently being encoded, are passed to transform block 1510 where they too are subjected to a forward transform (e.g DCT). The resulting transform coefficients c_{orig} are then passed to summer 1520 which forms a difference between c_{orig} and c_{pred} to generate prediction error transform coefficients c_{err} according to the relationship:

$$c_{err} = c_{orig} - c_{pred}.$$

The prediction error transform coefficients are supplied to quantisation block 1525 where they are quantised using quantisation parameter PQP to form quantised prediction error transform coefficients l_{err} which are, in turn passed to multiplexer 1540.

If the current block is being encoded in SP-format, multiplexer 1540 also receives information relating to the motion vectors used in formation of the motion compensated prediction P(x,y) for the SP-coded block. If the current block is being encoded in SI-format information concerning the intra prediction mode used to form the prediction P(x,y) for the SI-coded block is also passed to the multiplexer. Advantageously, multiplexer 1540 applies appropriate variable length coding (VLC) to the quantised prediction error transform coefficients l_{err} and to the motion vector or intra prediction mode information and forms a bit-stream for transmission to a corresponding decoder by multiplexing together the various forms of information.

The quantised prediction error transform coefficients l_{err} are passed from quantisation block 1525 to inverse quantisation block 1530 where they are inverse quantised using quantisation parameter PQP to form inverse quantised prediction error transform coefficients d_{err}. The inverse quantised prediction error transform coefficients d_{err} are next passed to summer 1540 where they are combined with the transform coefficients c_{pred} produced from the predicted pixel values P(x,y) for the current block. More specifically, summer 1540 adds transform coefficients c_{pred} and inverse quantised prediction error transform coefficients d_{err} to form reconstructed transform coefficients c_{rec} according to the relationship:

$$c_{rec} = c_{pred} + d_{err}$$

The reconstructed transform coefficients c_{rec} are then passed to quantisation block 1550 where they are quantised using quantisation parameter SPQP to produce quantised reconstructed transform coefficients l_{rec}. It should be noted that quantisation parameter SPQP used to quantise the reconstructed transform coefficients is not necessarily the same as the quantisation parameter PQP used to quantise the prediction error transform coefficients c_{err} in quantisation block 1525. In particular, a finer quantisation can be applied to reconstructed transform coefficients c_{rec} and a courser quantisation can

be applied to prediction error coefficients c_{err}. This ultimately results in a smaller reconstruction error (distortion) when a decoded image is formed in the decoder.

Quantised reconstructed transform coefficients l_{rec} are next supplied to inverse quantisation block 1560 where they are inverse quantised using quantisation parameter SPQP to form inverse quantised reconstructed transform coefficients d_{rec}. The inverse quantised reconstructed transform coefficients d_{rec} are then passed to inverse transform block 1565 where they are subjected to an inverse transform operation e.g. an Inverse Discrete Cosine Transform (IDCT). As a result of this operation, a block of reconstructed pixel values $l_c(x,y)$ is formed for the image block in question. The block of reconstructed pixel values $l_c(x,y)$ is subsequently stored in frame memory 1570. As subsequent blocks of the frame being encoded in S-frame format undergo the previously described encoding and local decoding operations, a decoded version of the current frame is progressively assembled in frame memory 1570 from where it can be accessed and used in intra prediction of subsequent blocks of the same frame or in inter (motion compensated) prediction of subsequent frames in the video sequence.

Operation of an S-frame decoder 1300 according to the second embodiment of the invention will now be described with reference to Figure 13. The bit-stream generated by the S-frame encoder 1500 according to the second embodiment of the invention and described above in connection with Figure 15 is received by decoder 1300 and is demultiplexed into its constituent parts. The decoder reconstructs a decoded version of the S-frame on a block-by-block basis. As previously described, an S-frame generally comprises both SP-coded and SI-coded image blocks. For SP-coded image blocks, the information in the received bit-stream comprises VLC encoded motion vector information and VLC encoded quantised prediction error transform coefficients l_{err}. For image blocks encoded in SI-format, the information in the received bit-stream comprises VLC encoded information relating to the intra prediction mode used to form the intra

prediction for the SI-coded block, as well as VLC encoded quantised prediction error transform coefficients I_{err}.

When decoding an SP-coded image block, demultiplexer 1310 first applies appropriate variable length decoding (VLD) to the received bit-stream to recover the motion vector information and quantised prediction error coefficients I_{err}. It then separates the motion vector information from the quantised prediction error coefficients I_{err}. The motion vector information is supplied to motion compensated prediction block 1370 and the quantised prediction error coefficients I_{err} recovered from the received bit-stream are applied to inverse quantisation block 1320. The motion vector information recovered from the received bit-stream is used in motion compensated prediction block 1370 together with pixel values of a previously reconstructed frame held in frame memory 1360 to form a prediction P(x,y) for the current block being decoded, in a manner analogous to that employed in encoder 1500.

When decoding an SI-coded image block, demultiplexer 1310 applies appropriate variable length decoding to the received intra prediction mode information and the quantised prediction error transform coefficients I_{err}. The intra prediction mode information is then separated from the quantised prediction error transform coefficients I_{err} and is supplied to intra prediction block 1380. The quantised prediction error transform coefficients I_{err} are supplied to inverse quantisation block 1320. The intra prediction mode information recovered from the received bit-stream is used in intra prediction block 1380 in conjunction with previously decoded pixel values of the current frame held in frame memory 1360 to form a prediction P(x,y) for the current block being decoded. Again, the intra prediction process performed in decoder 1200 is analogous to that performed in the corresponding encoder 1500 previously described.

For both SP- and SI- coded image blocks the quantised prediction error transform coefficients I_{err} recovered from the received bit-stream are inverse quantised in inverse quantisation block 1320 using quantisation parameter PQP to form inverse quantised prediction error transform

coefficients d_{err}. The inverse quantised prediction error transform coefficients d_{err} are applied to one input of summer 1325.

Once a prediction P(x,y) for the current block of the frame being decoded has been formed, either by motion compensated prediction in motion compensated prediction block 1370 or by intra prediction in intra prediction block 1380, switch 1385 is operated as appropriate to supply the predicted pixel values P(x,y) to transform block 1390. Here a forward transform, for example a Discrete Cosine Transform (DCT) is applied to the predicted block of pixel values P(x,y) to form transform coefficients c_{pred}. Transform coefficients c_{pred} are then supplied to the second input of summer 1325 where they are combined with the inverse quantised prediction error transform coefficients received from inverse quantisation block 1320, to form reconstructed transform coefficients c_{rec}. More specifically, the reconstructed transform coefficients are determined by adding together transform coefficients c_{pred} and inverse quantised prediction error transform coefficients d_{err} according to the relationship:

$$c_{rec} = c_{pred} + d_{err}.$$

The reconstructed transform coefficients c_{rec} are then passed to quantisation block 1330 where they are quantised to produce quantised reconstructed transform coefficients l_{rec} using quantisation parameter SPQP. The quantised reconstructed transform coefficients l_{rec} are, in turn, supplied to inverse quantisation block 1340 where they are inverse quantised using quantisation parameter SPQP to form inverse quantised reconstructed transform coefficients d_{rec}. The inverse quantised reconstructed transform coefficients d_{rec} are then passed to inverse transform block 1350 where they are subjected to an inverse transform operation, for example an Inverse Discrete Cosine Transform (IDCT). As a result of the inverse transformation applied in inverse transform block 1350, a block of reconstructed image pixels $l_c(x,y)$ is formed for the image block in question. The block of reconstructed pixels $l_c(x,y)$ is supplied to the video output of the decoder and to frame memory 1360, where the pixels are stored. As subsequent blocks of

the S-frame undergo the decoding operations described above, a decoded version of the current frame is progressively assembled in frame memory 1360. From there in can be accessed and used in the intra prediction of subsequent blocks of the same frame or in inter (motion compensated) prediction of subsequent frames in the video sequence.

In Fig. 16, an encoder according to a third embodiment of the invention is shown. In this embodiment, the transform coefficients c_{pred} undergo quantisation and inverse quantisation using the same quantisation parameter SPQP both in the encoder section (blocks 1625 and 1630) and in the decoder section (blocks 1692 and 1694). Therefore, the encoder does not introduce any additional quantisation error to the prediction loop, and error build-up in the prediction loop is thus effectively prevented. The blocks 1610, 1620, 1625, 1630, 1640, 1650, 1660, 1665, 1670, 1675, 1680, 1685, 1690 have similar functionality to the blocks 1510, 1520, 1525, 1530, 1540, 1550, 1560, 1565, 1570, 1575, 1580, 1585, 1590 presented in Fig. 15, respectively.

In Fig. 6, a decoder 600 according to an advantageous embodiment of the invention is described. Most of the elements of the decoder 600 are the same than of the decoder 1200 presented in Fig. 12. The operational blocks of the decoder 600 are arranged to decode the prediction blocks of the frames, wherein no switching means are not shown in Fig. 6. Other blocks 610, 615, 620, 630, 640, 650, 660 and 670 have similar functionality to the blocks 1210, 1220, 1230, 1240, 1250, 1260, 1290 and 1295 presented in Fig. 12, respectively.

In Fig. 9, a decoder 600 according to another advantageous embodiment of the invention is described. The decoder 600 presented in Fig. 9 is a modification from the decoder 600 shown in Fig. 6. The difference between the decoder presented in Fig. 9 and the decoder presented in Fig. 12 is that a normalisation block 680 is inserted between the demultiplexer 610 and one input of summing element 615. Other blocks 610, 615, 620, 630, 640, 650, 660 and 670 have similar

functionality to the blocks 1210, 1220, 1230, 1240, 1250, 1260, 1290 and 1295 presented in Fig. 12, respectively.

In Fig. 10, a decoder 600 according to yet another advantageous embodiment of the invention is described. Most of the elements of the decoder 600 are the same than of the decoder 1300 presented in Fig. 13. The operational blocks of the decoder 600 are arranged to decode the prediction blocks of the frames, wherein no switching means are not shown in Fig. 10. The other difference between the decoder presented in Fig. 13 and the decoder presented in Fig. 10 is that a normalisation block 680 is used instead of inverse quantisation block 1230. Other blocks 610, 615, 620, 630, 640, 650, 660 and 670 have similar functionality to the blocks 1310, 1325, 1330, 1340, 1350, 1360, 1370 and 1390 presented in Fig. 13, respectively.

The encoding of a video frame can be performed block-by-block basis such that in the same encoded video frame there can exist differently encoded areas. For example, some parts of the frame can be inter coded and some other parts of the frame can be intra coded. The above procedures are applied to each part of the frame appropriately, according to the encoding procedure of the part in question.

In addition to the transmission network, the request for the change of the bit stream transmission properties may also be originated by other parts of the transmission system. For example, the receiver may request the transmitting server to change the parameters for some reason. This request is delivered to the transmitting server *e.g.* via the transmission network.

Although H.26L is used as an example of a standard, embodiments of the present invention and any variations and modifications are deemed to be within the scope of the present invention.

Bit stream switching is not the only application in which the present invention can be applied. If one of the bit streams has a lower temporal resolution, e.g. 1 frame/sec, this bit stream can be used to provide fast-

forward functionality. Specifically, decoding from the bit stream with a lower temporal resolution and then switching to the bit stream with a normal frame rate would provide such functionality. Fig. 8 depicts two bit streams the second of which comprises only S-frames predicted from each other at intervals greater than the frame repetition interval of the first bit-stream. Furthermore, "Fast Forward" can start and stop at any location in the bit-stream. In the following, some other applications of the present invention are described.

Splicing and Random Access

The bit stream-switching example discussed earlier considered bit streams belonging to the same sequence of images. However, this is not necessarily the case in all situations where bit stream switching is needed. Examples include: switching between bit streams arriving from different cameras capturing the same event but from different perspectives, or cameras placed around a building for surveillance; switching to local/national programming or insertion of commercials in a television broadcast; video bridging, etc. The general term for the process of concatenating encoded bit streams is splicing.

When switching occurs between bit streams that belong to different sequence of images, this only affects the encoding of the S-frames used to switch between bit streams, i.e., the secondary S-frame S_{12} in Fig. 5. Specifically, the use of motion-compensated prediction of frames in one sequence of images using reference frames from a different sequence of images is not as efficient as when both bit streams belong to the same sequence of images. In this case, it is most likely that spatial prediction of the secondary S-frames frames would be more efficient. This is illustrated in Fig. 7 where the switching frame is an SI-frame using only spatial prediction, which identically reconstructs the corresponding SP-frame S_2. This method can be used as a random access mechanism into a bit stream and has further implications in error recovery and resilience as described in the following.

Error recovery

Multiple representations of a single frame in the form of S-frames predicted from different reference frames, e.g., predicted from the immediately preceding reconstructed frame and a reconstructed frame further back in time can be used to increase error resilience of an encoded video sequence and/or to improve recovery from errors in the bit-stream. This is illustrated in Fig. 14. In a situation where a packet loss occurs during streaming of a pre-encoded bit stream and a frame or slice is lost, the receiver informs the transmitter of the lost frame/slice and the transmitter responds by sending one of the alternative representations of the next S-frame. The alternative representation, *e.g.* frame S_{12} in Fig. 14, uses reference frames that have already been correctly received by the receiver. For slice based packetization and delivery, the sender could further estimate the slices that would be affected by such a slice/frame loss and update only those slices in the next S-frame with their alternative representations.

Similarly, and as mentioned earlier in the discussion of splicing, the secondary representation of the S-frame can be generated without using any reference frames, i.e., an SI_2-frame such as that shown in Fig. 14. In this case, the sender would send the secondary SI-frame, i.e., SI_2 instead of S_2 to stop error propagation. This approach could also be extended in a straight forward manner to slice-based encoding/packetization. More specifically, the server sends the slices in the next S-frame, which would be affected by the packet loss, from SI-frame.

Error resilience

The encoding of a video frame can be performed block-by-block basis such that in the same encoded video frame there can exist differently encoded areas. For example, some parts of the frame can be inter coded and some other parts of the frame can be intra coded. And as already discussed, the intra-block coding, since it doesn't employ any

temporal correlation, stops any error propagation which might be started due to transport impairments.

In lossy transport networks, an intra macroblock refresh strategy can provide significant error resiliency/recovery performance. In an interactive client/server scenario, the encoder on the server side decides to encode the frames/macroblocks either based on specific feedback received from the client, e.g., the exact location of the lost/corrupted frame/slice/macro block, or based on the expected network conditions calculated through negotiation or the measured network conditions. This kind of intra-macroblock update strategy improves the quality of the received video by providing error resiliency and error recovery. The optimal intra-macroblock update refresh rate, i.e., the frequency at which macroblocks are intra-coded, depends on the transport channel conditions, e.g., packet loss and/or bit error rate. However, when already encoded bit streams are sent, which is the case in typical streaming applications, the above strategy cannot be applied directly. Either the sequence needs to be encoded with the worst-case expected network conditions or additional error resiliency/recovery mechanisms are required.

From the earlier discussion concerning the use of S-frames in error recovery and splicing applications, it can be noted that S-frames or slices within them can easily be represented as SI-frames/slices that do not use any reference frames while still leading to identical reconstruction of the S-frame. This feature can be exploited in the adaptive intra refresh mechanism discussed above. First, a sequence of images is encoded with some predefined ratio of S-macroblocks. Then during the transmission, some of the S-macroblocks are sent in their secondary representation that is as SI-macro blocks. The number of S-macroblocks to be sent in SI representation can be calculated in a manner similar to the method used in the real-time encoding/delivery approach previously described.

Video Redundancy Coding

S-frames have other uses in applications in which they do not act as replacements of I-frames. Video Redundancy Coding (VRC) can be given as an example. The principle of the VRC method is to divide a sequence of pictures into two or more threads in such a way that all pictures in the sequence are assigned to one of the threads in a round-robin fashion. Each thread is coded independently. At regular intervals, all threads converge into a so-called sync frame. From this sync frame, a new thread series is started. If one of the threads is damaged e.g. because of a packet loss, the remaining threads typically stay intact and can be used to predict the next sync frame. It is possible to continue the decoding of the damaged thread, which leads to slight degradation in picture quality, or to stop the decoding of the damaged thread, which leads to a drop in the frame rate. Sync frames are always predicted from one of the undamaged threads. This means that the number of transmitted I-frames can be kept small, because there is no need for complete re-synchronization. For the sync frame more than one representation (P-frame) is sent, each one using a reference frame from a different thread. Due to the use of P-frames these representations are not identical. Therefore mismatch is introduced when some of the representations cannot be decoded and their counterparts are used when decoding the following threads. The use of S-frames as sync frames eliminates this problem.

It is obvious that the present invention is not limited to the above described embodiments but it can be modified within the scope of the appended claims.

Claims:

1. A method for transmitting video information, in which at least a first bit-stream (510) and a second bit-stream are formed from the video information, the first bit-stream (510) comprises a first set of frames comprising at least one video frame, and the second bit-stream (520) comprises a second set (520) of frames comprising at least one predictive video frame (524), at least partly different encoding parameters are used with encoding of the frames of said first bit-stream (510) and said second bit-stream (520), at least one frame of said first bit-stream (510) is being transmitted, and the transmission is switched over from said first bit-stream (510) to said second bit-stream (520), **characterized** in that in switching the transmission over from said first bit-stream (510) to said second bit-stream (520), the secondary bit-stream comprises at least one primary switching frame (523), a secondary switching frame (550) is transmitted, the secondary switching frame (550) having been encoded using the encoding parameters of the second bit-stream (520) and at least one reference frame from the first bit-stream (510), and said secondary switching frame (550) is used in place of the primary switching frame (523) as a reference frame to be used in the reconstruction of said at least one predictive video frame (524) of the second set of video frames.

2. The method according to claim 1, **characterized** in that the first bit-stream (510) comprises at least one primary switching frame (513).

3. The method according to claim 1 or 2, **characterized** in that the first bit-stream (510) comprises only one intra frame and one primary switching frame (513) to perform a changeover from one location of the video information to another location of the video information.

4. The method according to claim 1 or 2, **characterized** in that the first bit-stream (510) comprises only intra frames and primary switching frames (513) to perform fast forward operation to the video information.

5. The method according to any of the claims 2 to 4, **characterized** in that said primary switching frame (523) is a predictive video frame (524), in which prediction information comprises only intra prediction information.

6. The method according to any of the claims 1 to 5, **characterized** in that the primary switching frame (513) is formed so that transform coefficients c_{pred} are calculated and quantised to form quantised values of the transform coefficients l_{pred}, that quantised coefficients l_{err} for prediction error are defined, that dequantised reconstruction image coefficients d_{rec} are defined so that coefficients c_{rec} exist from which by quantisation and dequantisation, dequantised reconstruction image coefficients d_{rec} can be obtained, and that dequantised reconstruction image coefficients d_{rec} fulfil one of the following conditions:

$d_{rec} = d_{pred} + d_{err}$, or
$c_{rec} = c_{pred} + d_{err}$,

in which d_{err} are dequantised values of the prediction error, and
d_{pred} are the dequantised values of the quantised transform coefficients l_{pred}.

7. The method according to claim 6, **characterized** in that same quantisation parameters are used with quantisation of the transform coefficients l_{pred}, and with quantisation of prediction error.

8. The method according to claim 6, **characterized** in that different quantisation parameters are used with quantisation of the transform coefficients l_{pred}, than with quantisation of prediction error.

9. The method according to any of the claims 1 to 8, **characterized** in that said secondary switching frame (550) is used for recovering from transmission errors, wherein said primary switching frame (523) is a predictive video frame (524), in which prediction information comprises prediction information from previous video frames than the preceding frame of the predictive video frame (524).

10. The method according to any of the claims 1 to 8, **characterized** in that said secondary switching frame (550) is used for recovering from transmission errors, wherein said primary switching frame (523) is a predictive video frame (524), in which prediction information comprises only intra prediction information.

11. The method according to any of the claims 1 to 10, **characterized** in that both said at least one primary switching frame (523) and said secondary switching frame (550) produce the same reconstruction result of said at least one predictive video frame (524).

12. An encoder comprising means for forming at least a first bit-stream (510) and a second bit-stream from a video information, the first bit-stream comprises a first set of frames comprising at lest one video frame, and the second bit-stream comprises a second set (520) of frames comprising at least one predictive video frame (524), and means for using at least partly different encoding parameters with encoding of the frames of said first bit-stream (510) and said second bit-stream (520), means for transmitting at least one frame of said first bit-stream (510), and means for switching the transmission over from said first bit-stream (510) to said second bit-stream (520), **characterized** in that said means for switching the transmission over from said first bit-stream (510) to said second bit-stream (520) comprise means for encoding a secondary switching frame (550) using the encoding parameters of the second bit-stream (520) and reference frames from the first bit-stream (510) to enable to switch the transmission over from said first bit-stream (510) to said second bit-stream (510).

13. The encoder according to claim 12, **characterized** in that it comprises means (1670, 1675) for producing prediction information by using said reference frames, and means (1692, 1694) for performing quantisation and dequantisation to the prediction information.

14. A decoder for decoding a video information from a signal comprising frames from at least a first bit-stream (510) and a second bit-stream, which are formed from the video information, the first bit-stream comprises a first set of frames comprising at lest one video frame, and the second bit-stream comprises a second set (520) of frames comprising at least one predictive video frame (524), and at least partly different encoding parameters are used with encoding of the frames of said first bit-stream (510) and said second bit-stream (520), **characterized** in that the decoder comprises means for decoding a secondary switching frame (550), said secondary switching frame (550) having been encoded using the encoding parameters of the second bit-stream (520) and at least one reference frame from the first bit-stream (510), and added to said signal in place of a primary switching frame (523) as a reference frame to be used in the reconstruction of said at least one predictive video frame (524) of the second set of video frames, and that said means for decoding a secondary switching frame (550) comprise means for using decoding parameters of the second bit-stream (520) and reference frames from the first bit-stream (510).

15. The decoder according to claim 14, **characterized** in that the primary switching frame (513) comprises areas encoded by intra prediction using only spatial correlation, and areas encoded by inter prediction using motion compensation, wherein the decoder comprises means for using motion compensation information with reconstruction, means for using spatial correlation information with reconstruction, and switching means to perform the reconstruction of each area either by said means for using motion compensation information or by said means for using spatial correlation information, depending on the prediction method used by encoding of the respective area.

16. A signal representing encoded video information, and comprising frames from at least a first bit-stream (510) and a second bit-stream, which are formed from the video information, the first bit-stream comprises a first set of frames comprising at lest one video frame, and the second bit-stream comprises a second set (520) of frames

comprising at least one predictive video frame (524), at least partly different encoding parameters are used with encoding of the frames of said first bit-stream (510) and said second bit-stream (520), **characterized** in that in switching the transmission over from said first bit-stream (510) to said second bit-stream (520), the secondary bit-stream comprises at least one primary switching frame (523), said signal comprises a secondary switching frame (513, 523) having been encoded using the encoding parameters of the second bit-stream (520) and at least one reference frame from the first bit-stream (510), and said secondary switching frame (550) is in place of the primary switching frame (523) as a reference frame to be used in the reconstruction of said at least one predictive video frame (524) of the second set of video frames.

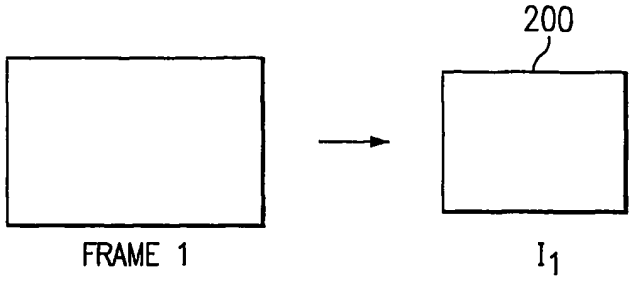


FIG. 1A
(PRIOR ART)

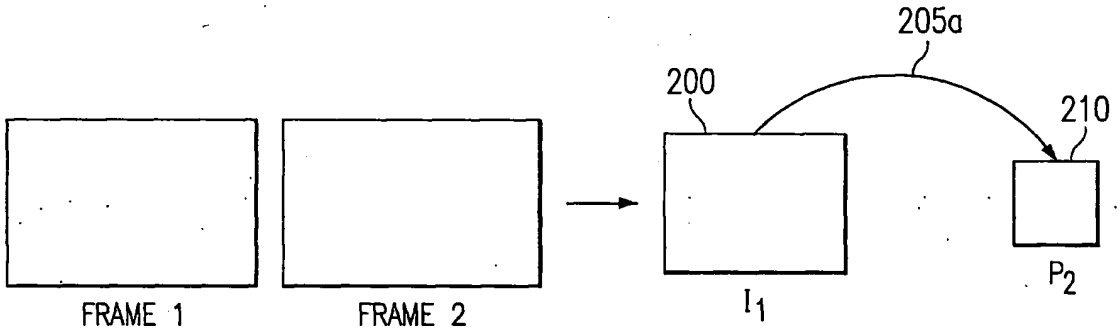


FIG. 1B
(PRIOR ART)

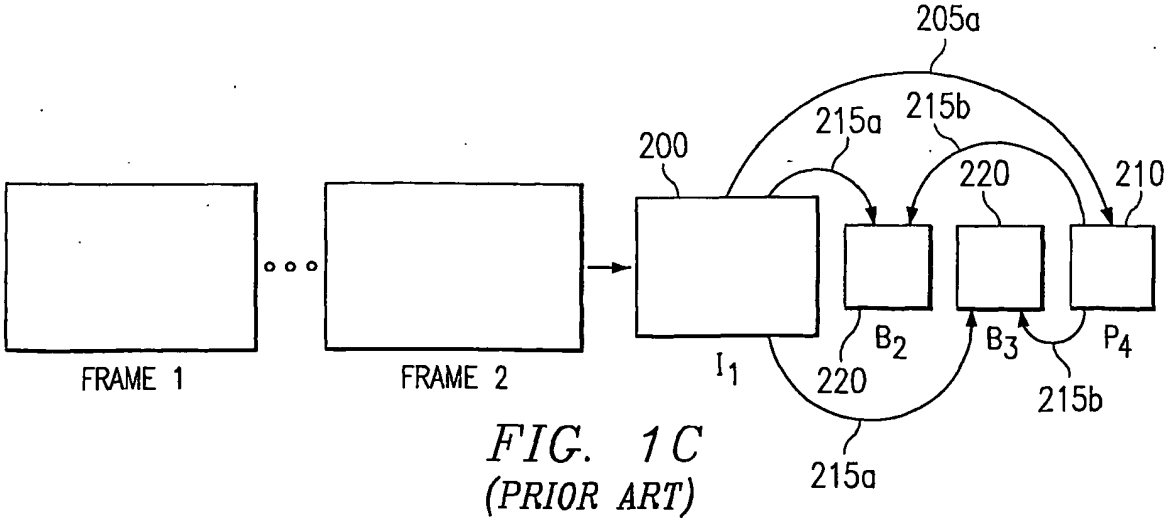


FIG. 1C
(PRIOR ART)

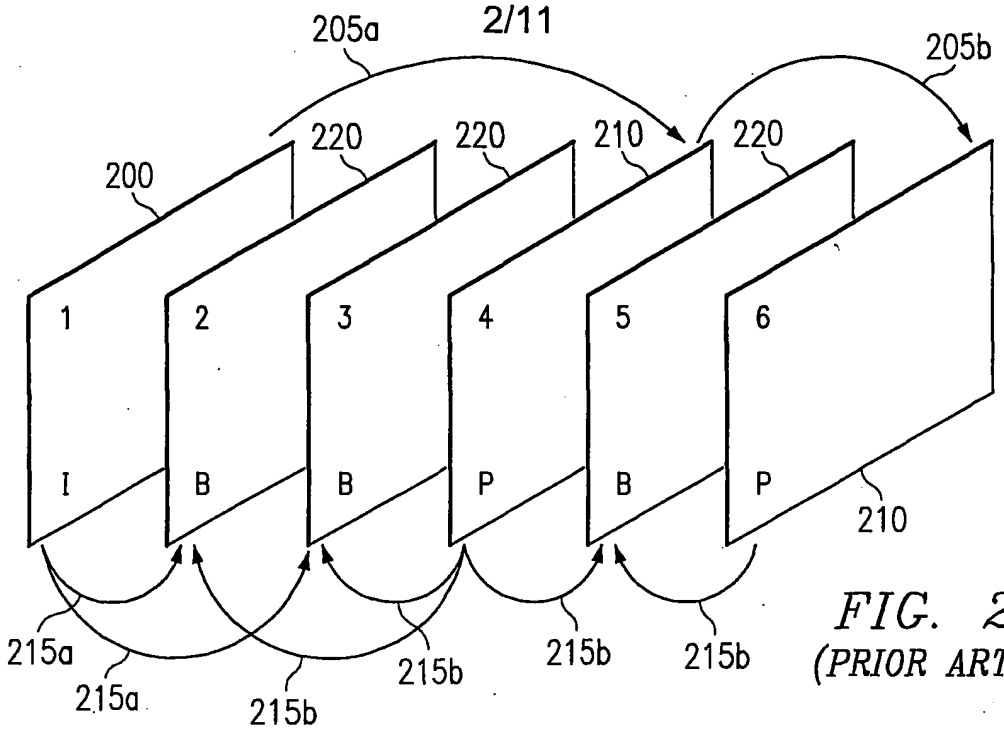


FIG. 2
(PRIOR ART)

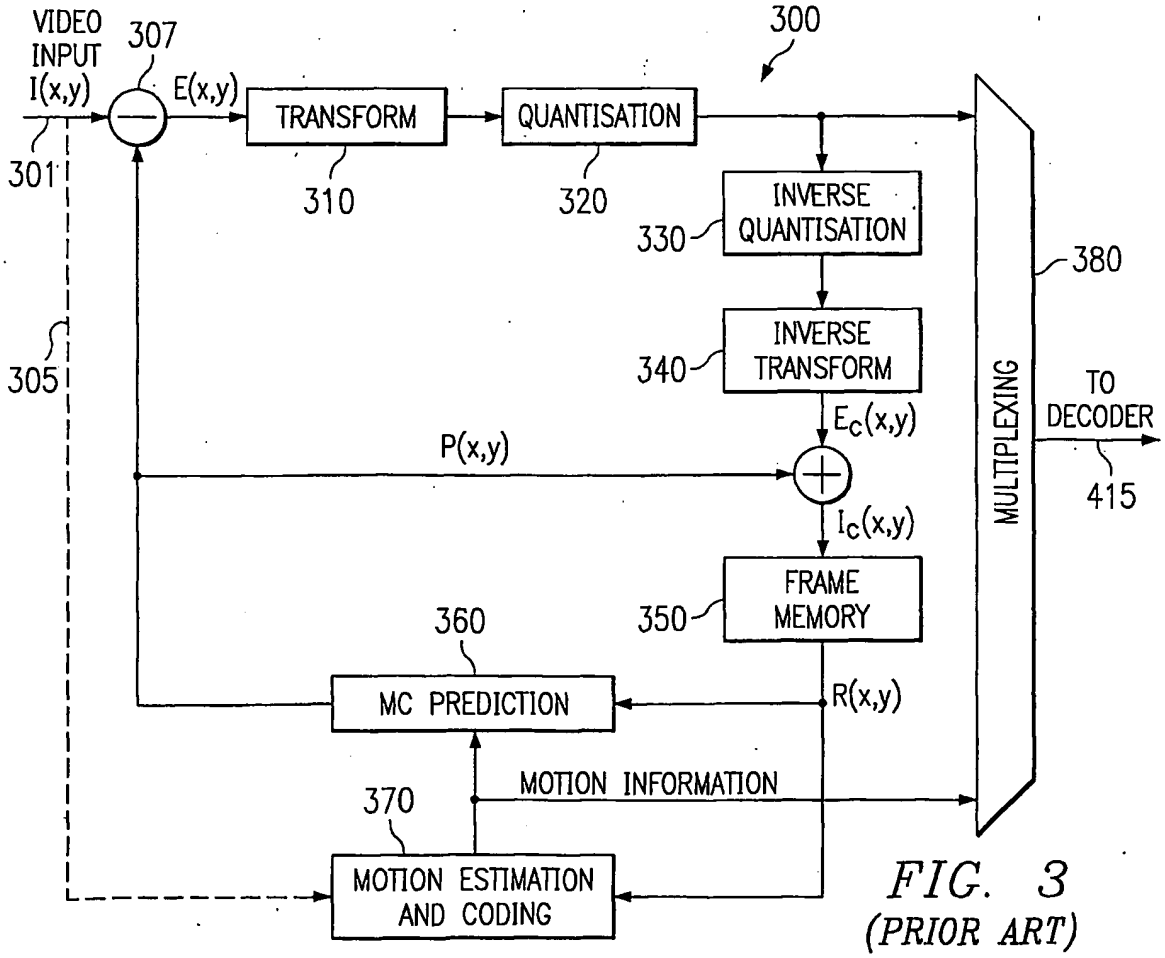


FIG. 3
(PRIOR ART)

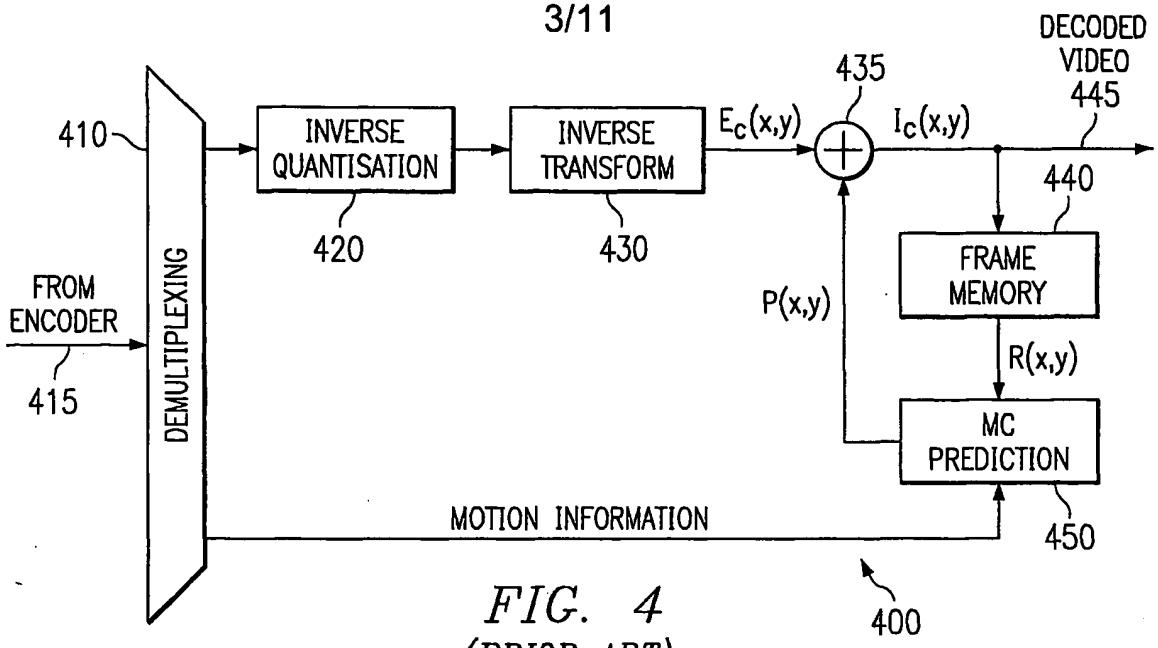


FIG. 4
(PRIOR ART)

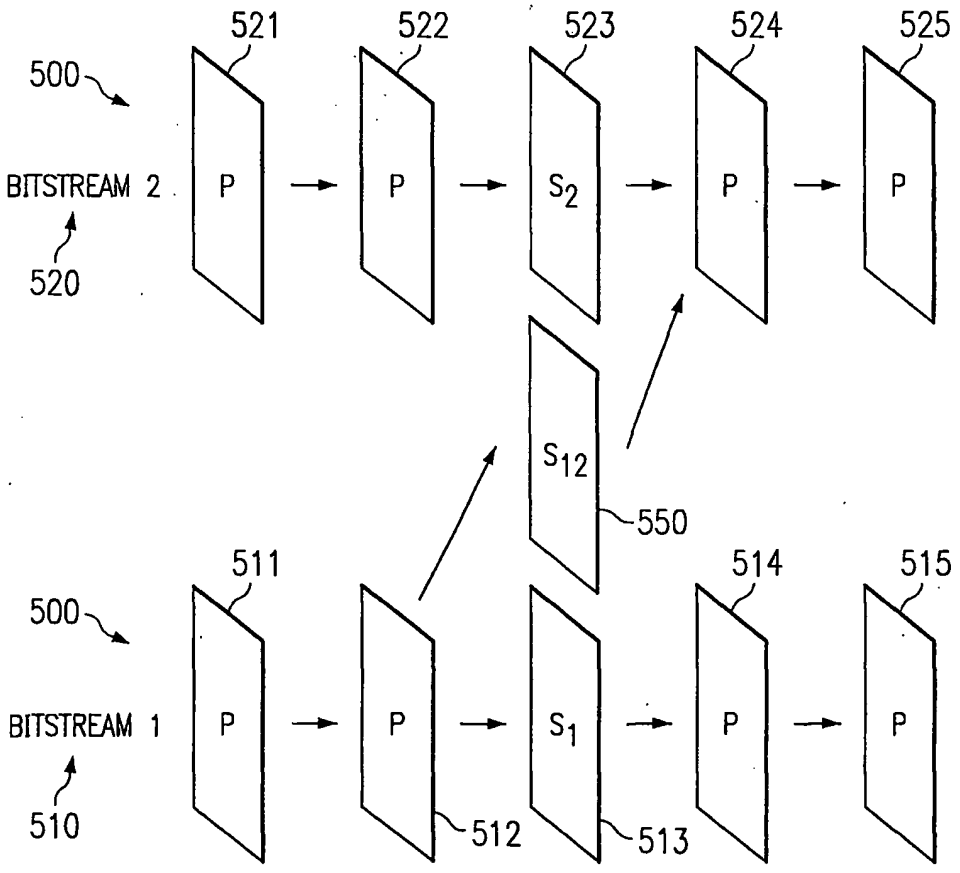


FIG. 5

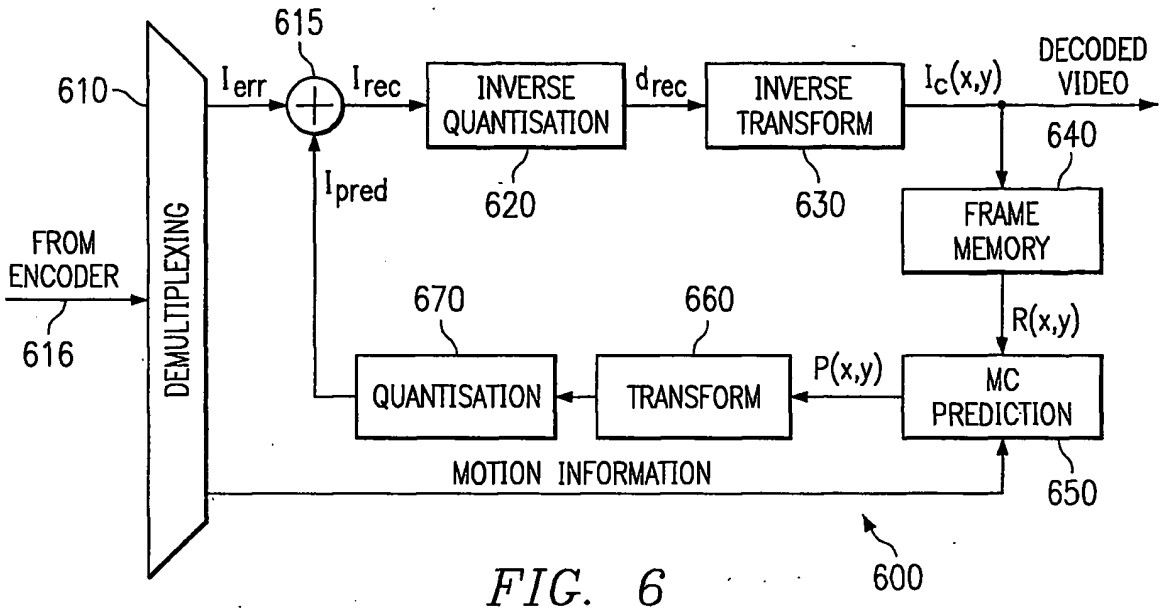

615
610
I_{err}
I_{rec}
INVERSE QUANTISATION
d_{rec}
INVERSE TRANSFORM
$I_C(x,y)$
DECODED VIDEO
640
I_{pred}
620
630
FRAME MEMORY
FROM ENCODER
616
DEMULTIPLEXING
670
660
R(x,y)
QUANTISATION
TRANSFORM
P(x,y)
MC PREDICTION
MOTION INFORMATION
650
600

FIG. 6

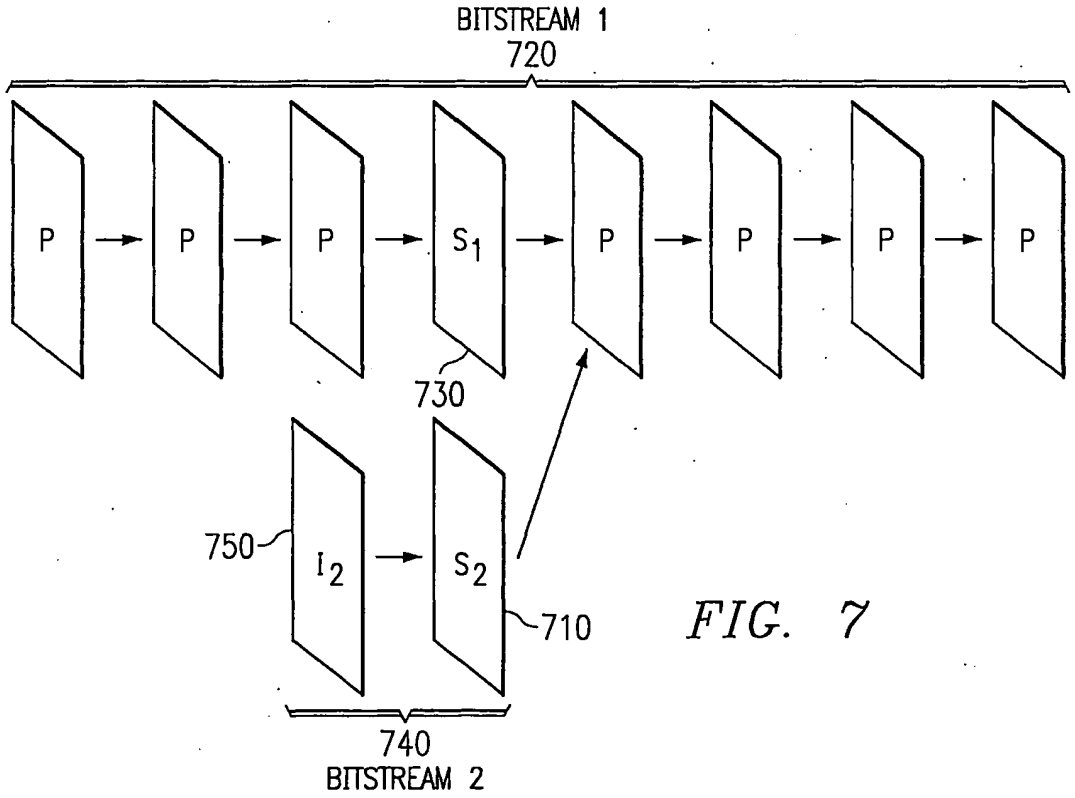

BITSTREAM 1
720
P
P
P
S_1
P
P
P
P
730
750
I_2
S_2
710
740
BITSTREAM 2

FIG. 7

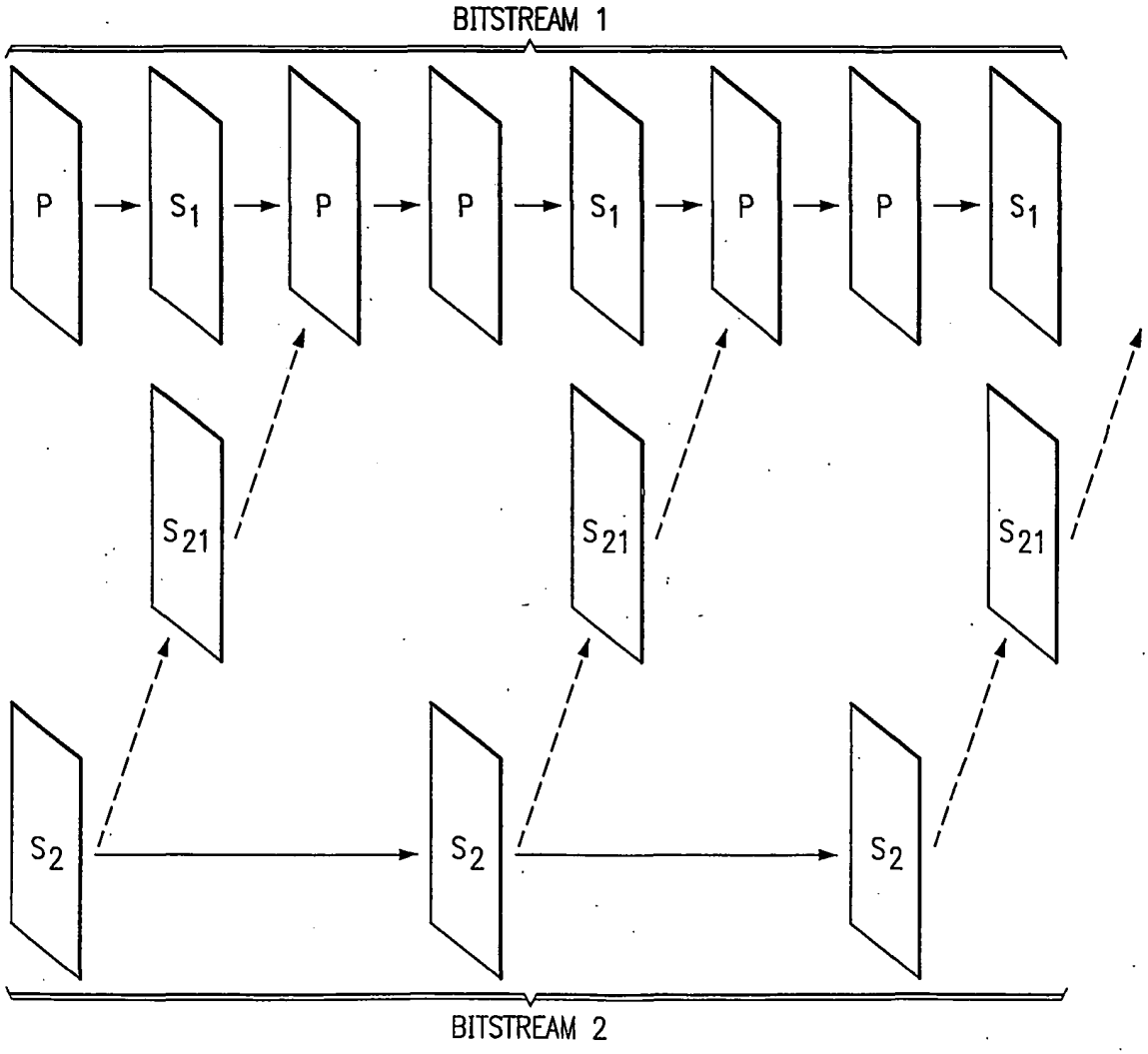


FIG. 8

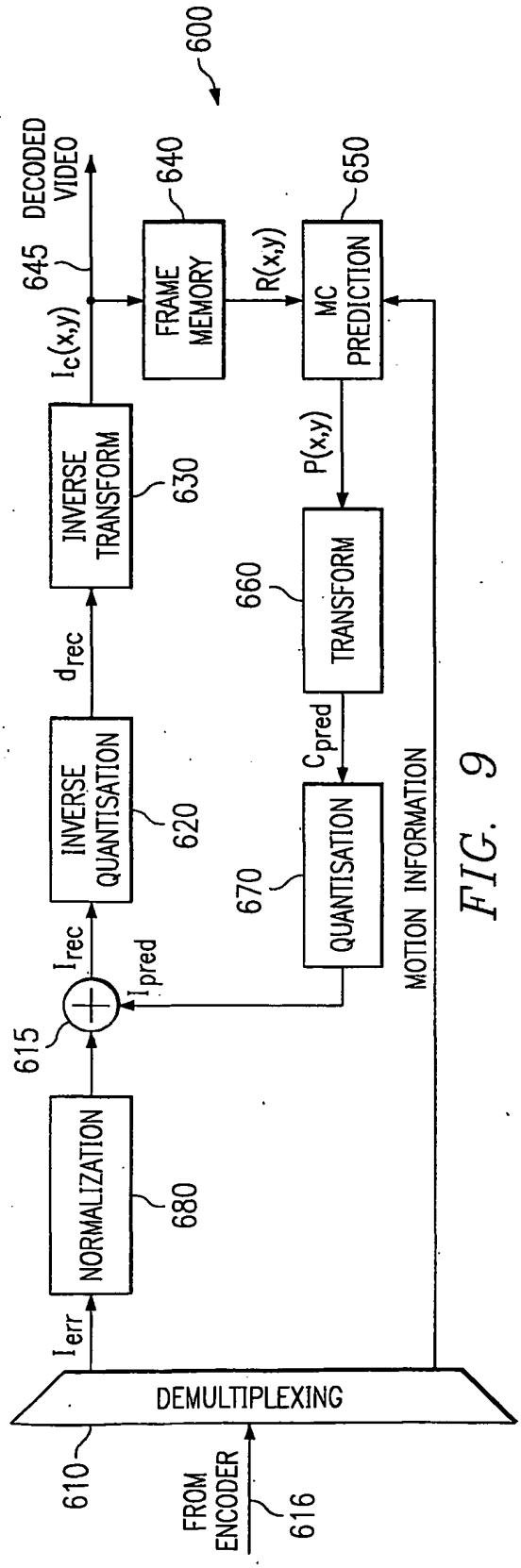

600
DECODED VIDEO
645
$I_c(x,y)$
INVERSE TRANSFORM
630
d_{rec}
INVERSE QUANTISATION
620
I_{rec}
615
I_{pred}
NORMALIZATION
680
I_{err}
DEMULTIPLEXING
610
FROM ENCODER
616
FRAME MEMORY
640
R(x,y)
MC PREDICTION
650
P(x,y)
660
TRANSFORM
C_{pred}
670
QUANTISATION
MOTION INFORMATION

FIG. 9

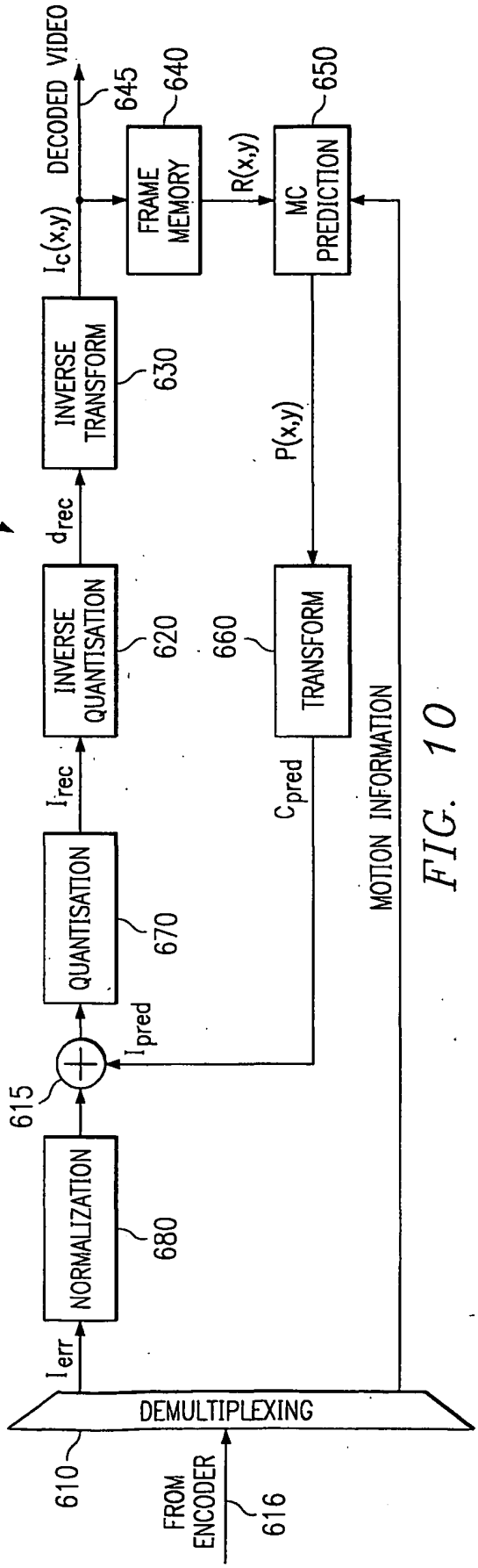

600
DECODED VIDEO
645
$I_c(x,y)$
INVERSE TRANSFORM
630
d_{rec}
INVERSE QUANTISATION
620
I_{rec}
615
I_{pred}
NORMALIZATION
680
I_{err}
DEMULTIPLEXING
610
FROM ENCODER
616
FRAME MEMORY
640
R(x,y)
MC PREDICTION
650
P(x,y)
660
TRANSFORM
C_{pred}
670
QUANTISATION
MOTION INFORMATION

FIG. 10

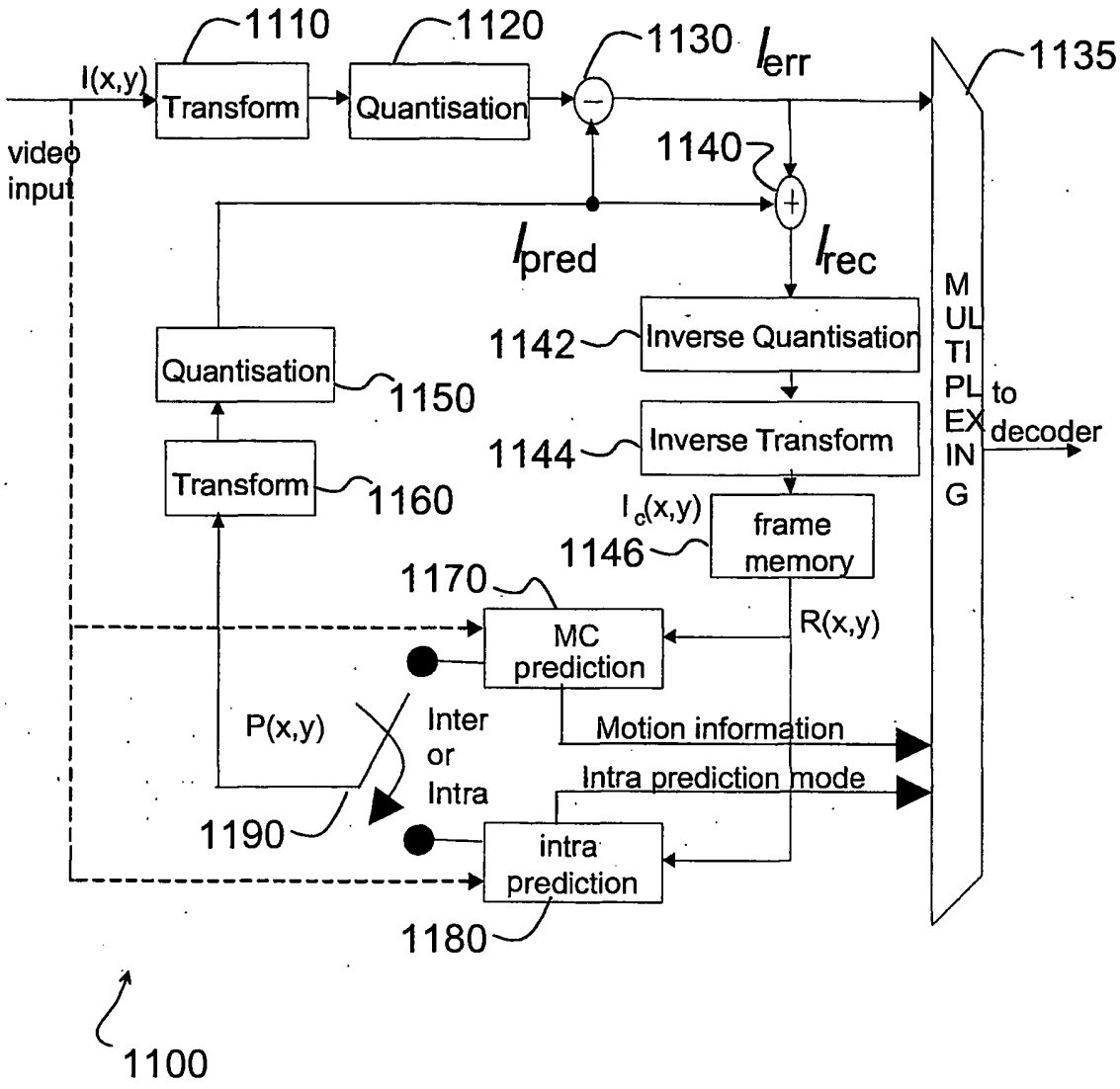

1110
1120
1130
I_{err}
1135
I(x,y)
Transform
Quantisation
video input
1140
I_{pred}
I_{rec}
MULTIPLEXING
to decoder
1142
Inverse Quantisation
Quantisation
1150
1144
Inverse Transform
Transform
1160
$I_c(x,y)$
frame memory
1146
1170
MC prediction
R(x,y)
P(x,y)
Inter or Intra
Motion information
Intra prediction mode
1190
intra prediction
1180
1100

FIG. 11

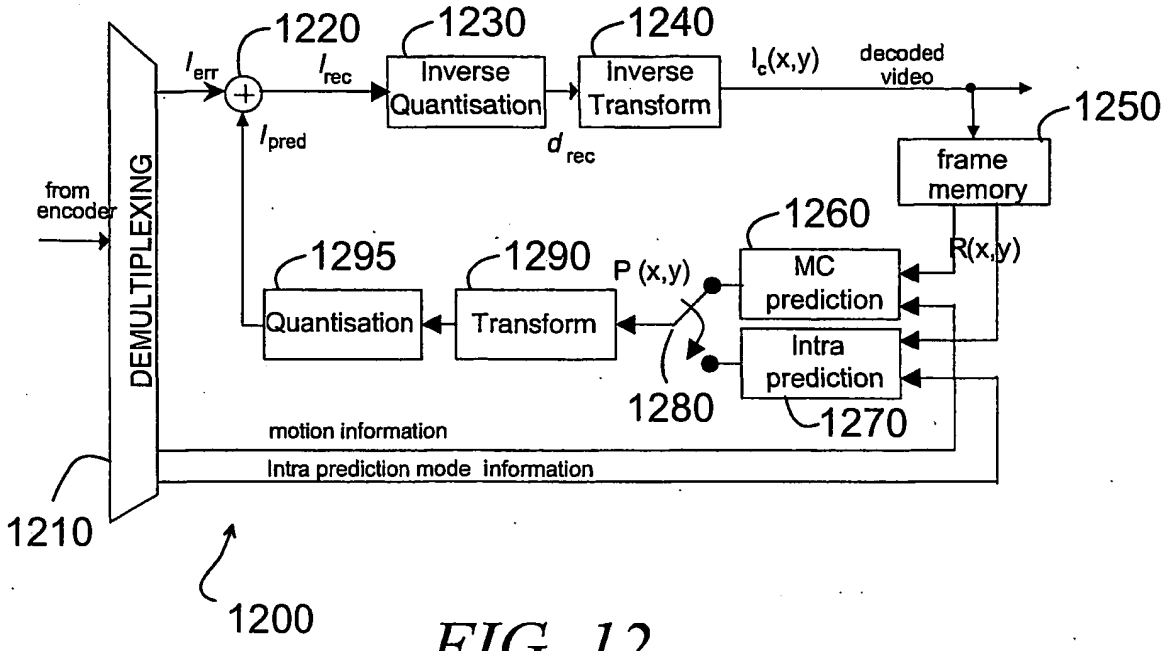

1220
1230
1240
1250
I_{err}
I_{rec}
Inverse Quantisation
Inverse Transform
$I_c(x,y)$
decoded video
I_{pred}
d_{rec}
frame memory
from encoder
DEMULTIPLEXING
1260
R(x,y)
1295
1290
P (x,y)
MC prediction
Quantisation
Transform
Intra prediction
1280
1270
motion information
Intra prediction mode information
1210
1200

FIG. 12

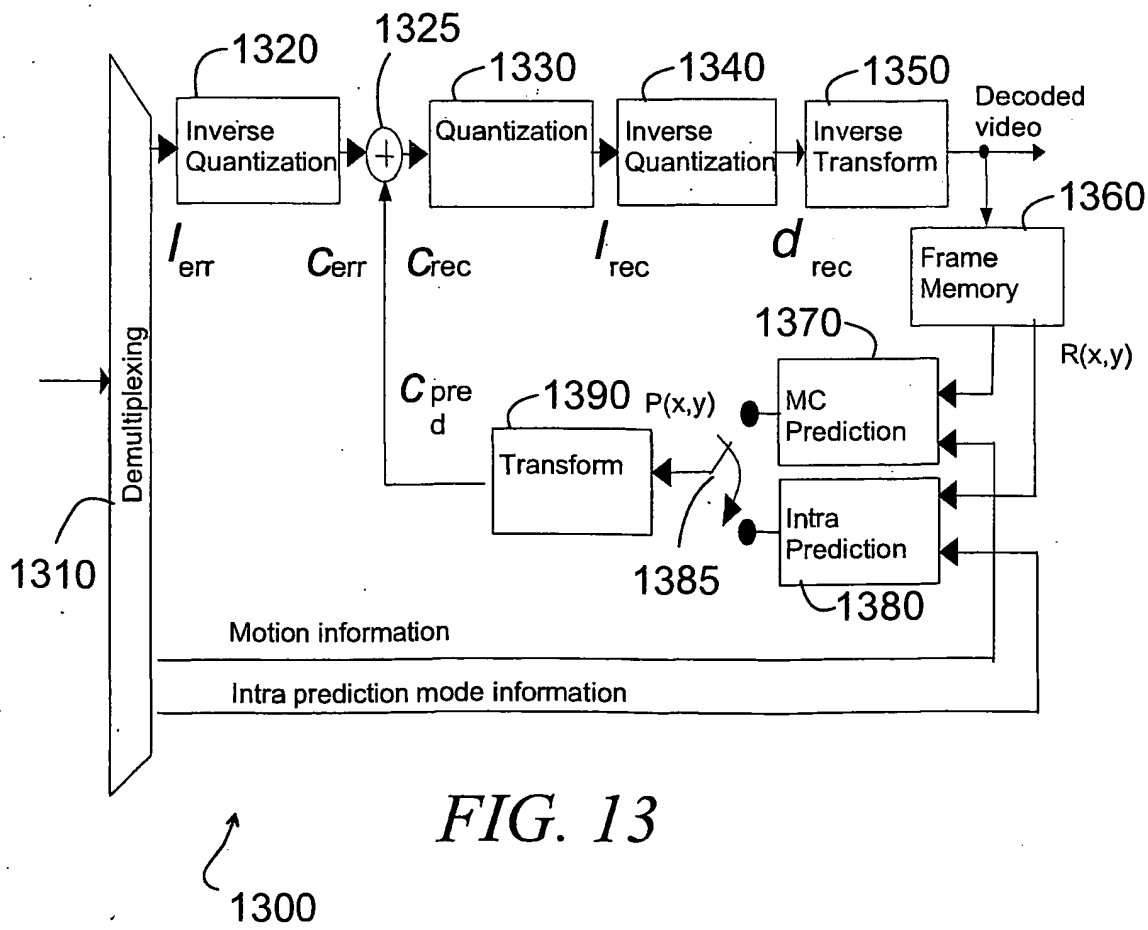

1320
1325
1330
1340
1350
Decoded video
Inverse Quantization
Quantization
Inverse Quantization
Inverse Transform
1360
I_{err}
C_{err}
C_{rec}
I_{rec}
d_{rec}
Frame Memory
1370
R(x,y)
C_{pred}
1390
P(x,y)
MC Prediction
Transform
Intra Prediction
1385
1380
Demultiplexing
1310
Motion information
Intra prediction mode information
1300

FIG. 13

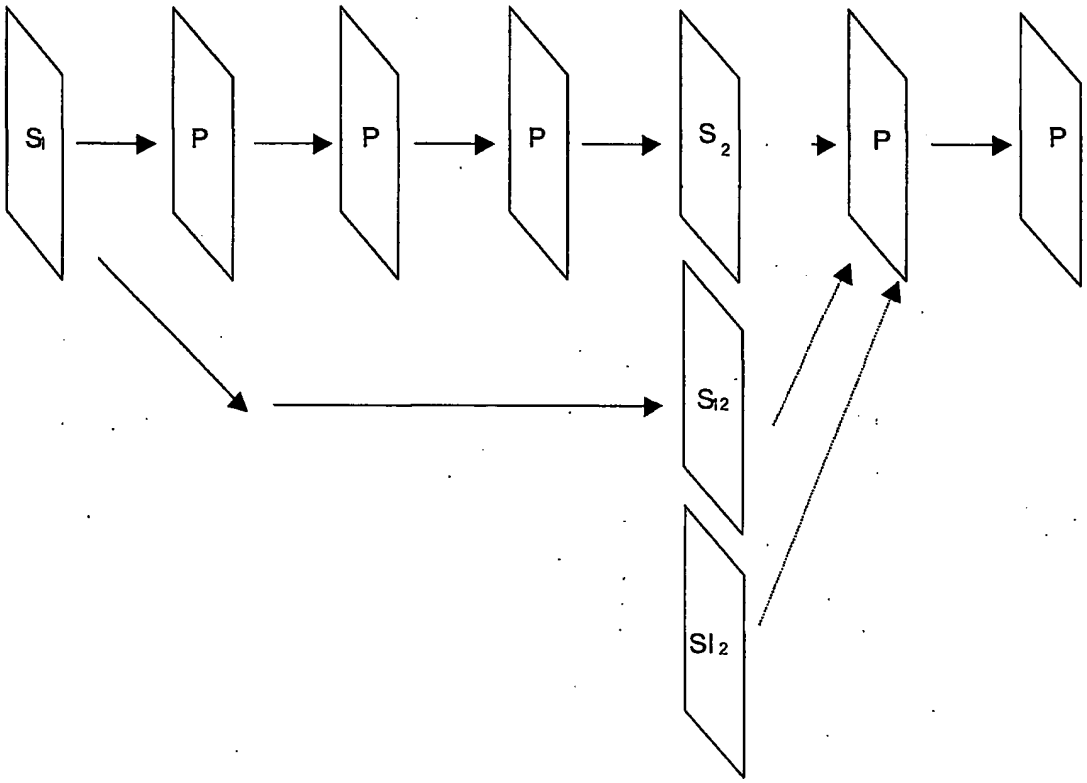


FIG. 14

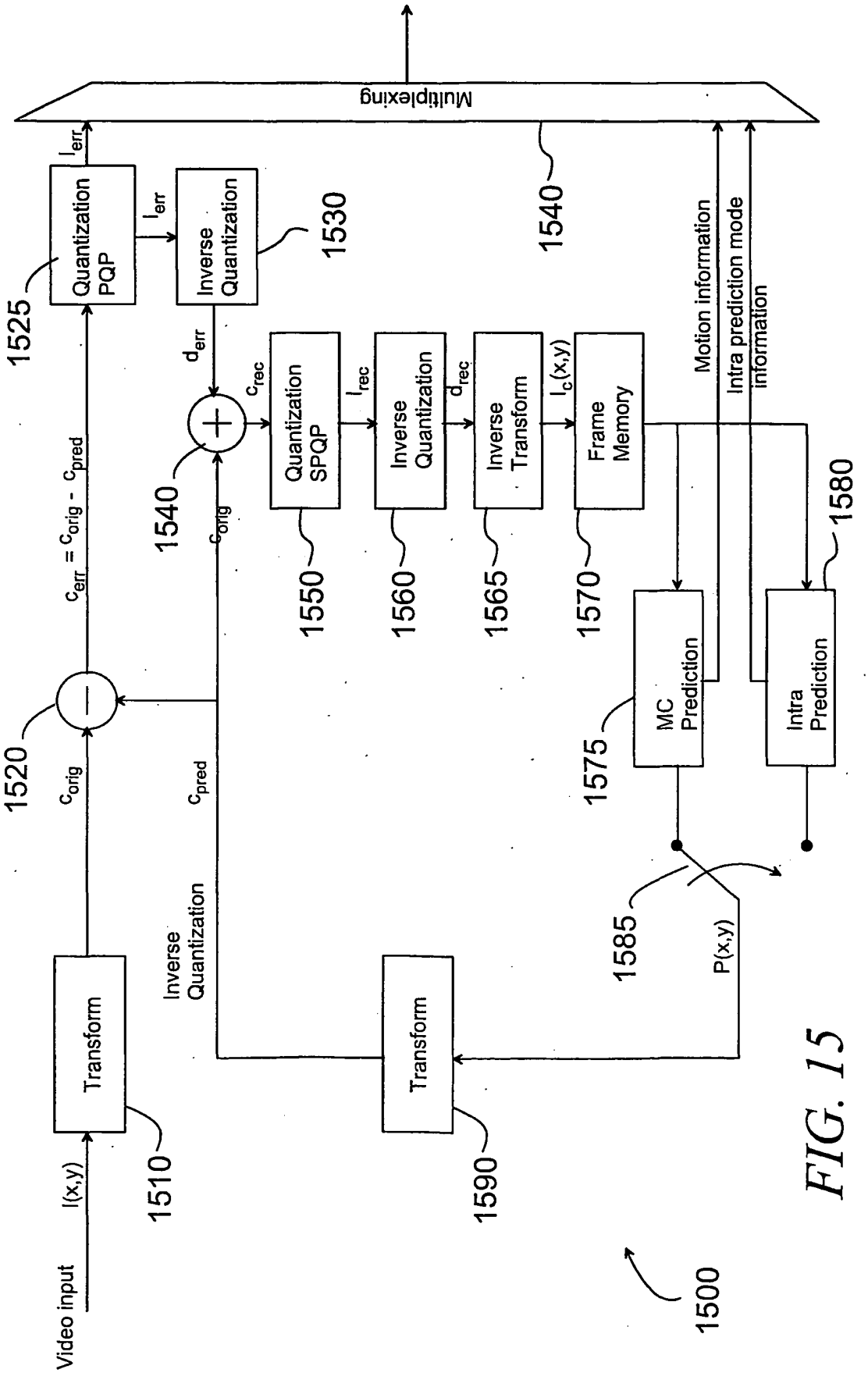

Multiplexing
1540
Quantization PQP
1525
Inverse Quantization
1530
Quantization SPQP
1550
Inverse Quantization
1560
Inverse Transform
1565
Frame Memory
1570
MC Prediction
1575
Intra Prediction
1580
Motion information
Intra prediction mode information
1520
1540
Transform
1510
Inverse Quantization
Transform
1590
1585
P(x,y)
Video input
I(x,y)
1500
I_{err}
d_{err}
c_{rec}
I_{rec}
d_{rec}
$I_c(x,y)$
$c_{err} = c_{orig} - c_{pred}$
c_{orig}
c_{pred}

FIG. 15

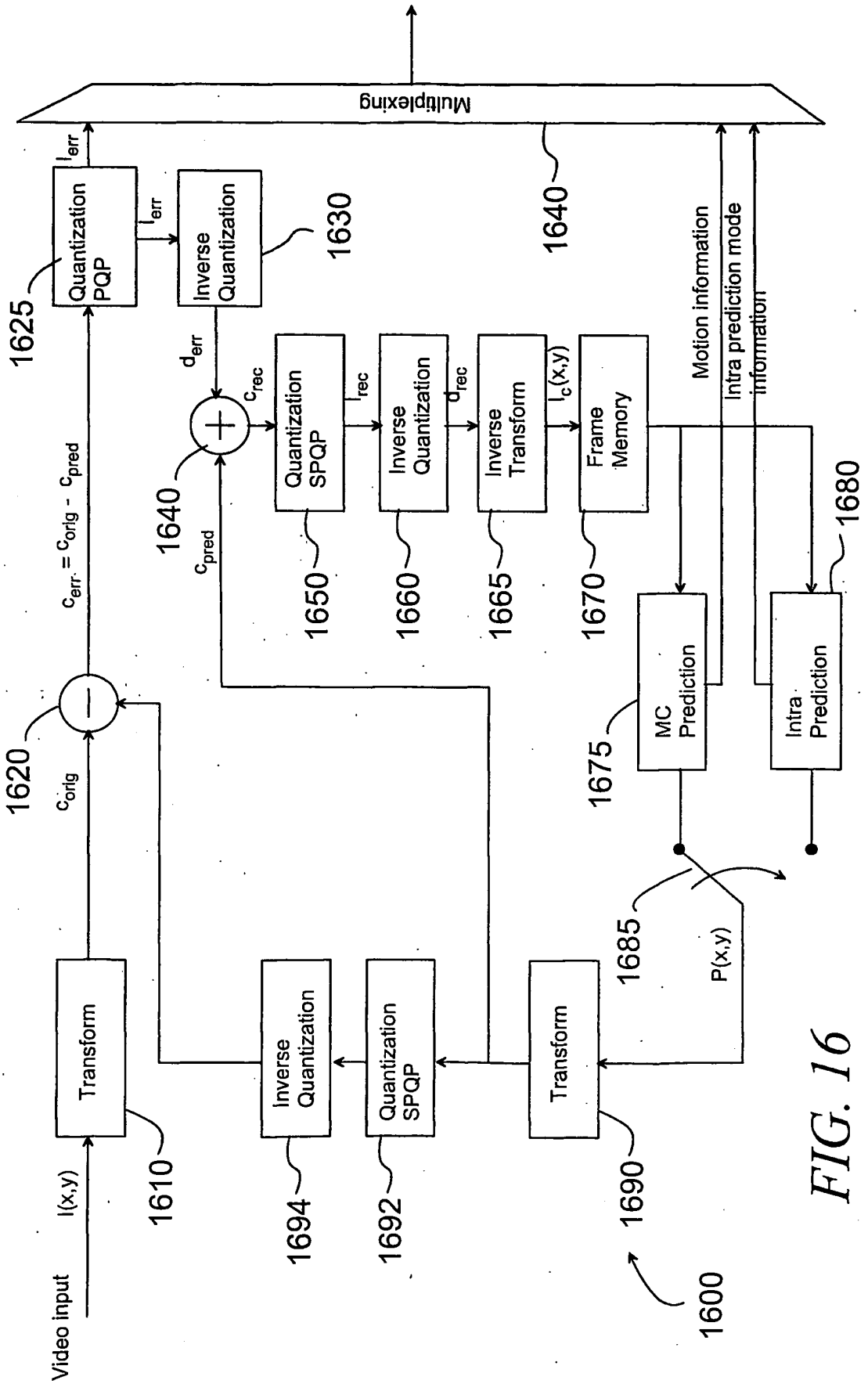

Multiplexing
1640
Motion information
Intra prediction mode information
1680
Frame Memory
1670
Inverse Transform
1665
Inverse Quantization
1660
Quantization SPQP
1650
Inverse Quantization
1630
Quantization PQP
1625
l_{err}
d_{err}
c_{rec}
l_{rec}
d_{rec}
$l_c(x,y)$
1640
c_{pred}
$c_{err} = c_{orig} - c_{pred}$
1620
c_{orig}
MC Prediction
1675
Intra Prediction
1685
P(x,y)
Transform
1690
Inverse Quantization
1694
Quantization SPQP
1692
Transform
1610
I(x,y)
Video input
1600

FIG. 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI 02/00004

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04N 7/26, H04N 7/58
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA, INSPEC, COMPENDEX, TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 1045589 A2 (SONY UK LTD), 18 October 2000 (18.10.00), abstract ---	1-16
P,A	EP 1079631 A1 (SONY UK LTD), 28 February 2001 (28.02.01), abstract ---	1-16
A	WO 9114340 A1 (THOMSON CONSUMER ELECTRONICS), 19 Sept 1991 (19.09.91), whole document ---	1-16
A	Patent Abstracts of Japan, abstract of JP 12-165816 A (SONY CORP), 16 June 2000 (16.06.00), abstract --- --------	1-16

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 April 2002	02-05-2002
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Jesper Bergstrand/LR Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT
Information on patent family members

28/01/02

International application No.
PCT/FI 02/00004

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1045589 A2	18/10/00	GB 9908809 D	00/00/00
		JP 2001054110 A	23/02/01
EP 1079631 A1	28/02/01	GB 2353653 A	28/02/01
		GB 9920274 D	00/00/00
		JP 2001119305 A	27/04/01
WO 9114340 A1	19/09/91	AT 154485 T	15/06/97
		AU 7480691 A	10/10/91
		DE 69126525 D,T	25/09/97
		EP 0519995 A,B	30/12/92
		JP 2891773 B	17/05/99
		JP 5505080 T	29/07/93
		KR 235815 B	15/12/99

Form PCT/ISA/210 (patent family annex) (July 1998)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 005 232 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
10.12.2003 Bulletin 2003/50

(43) Date of publication A2:
31.05.2000 Bulletin 2000/22

(21) Application number: **99120843.0**

(22) Date of filing: **25.10.1999**

(51) Int Cl.[7]: **H04N 7/26**, H04N 7/50, H04N 7/58, H04N 7/46

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **24.11.1998 US 198867**

(71) Applicant: **General Instrument Corporation**
Horsham, Pennsylvania 19044 (US)

(72) Inventors:
- **Wang, Limin**
  San Diego, California 92128 (US)
- **Luthra, Ajay**
  San Diego, California 92130 (US)
- **Eifrig, Robert O.**
  San Diego, California 92126 (US)

(74) Representative: **Hoeger, Stellrecht & Partner**
Uhlandstrasse 14 c
70182 Stuttgart (DE)

(54) Rate control for an MPEG transcoder without a priori knowledge of picture type

(57) A rate control system suitable for use with a digital video transcoder, such as one conforming to the MPEG standard. The proposed rate control system starts coding without a priori knowledge of the Group of Pictures (GOP) parameters, thereby avoiding a processing delay of about one GOP which would otherwise be incurred to extract the complete GOP structure information. In addition, the system avoids the need to store the data corresponding to the GOP, thereby reducing the memory required for transcoding. An initially allocated bit budget is gradually corrected as successive pictures are coded according to their picture types. Changes in the GOP structure of pre-compressed bitstreams can be addressed, for example, when switching channels or inserting commercials. Target rates with incorrect starting GOP parameters will converge within a few GOPs.

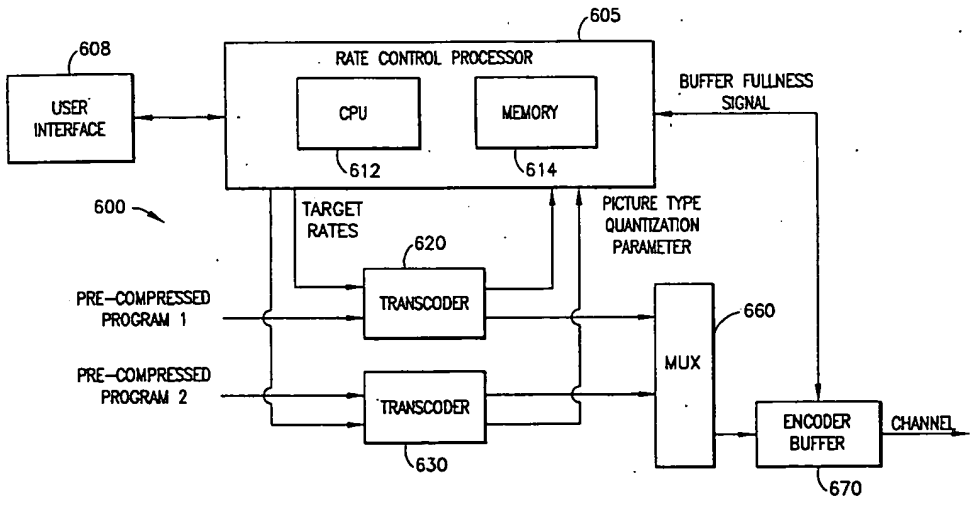


FIG.6

EP 1 005 232 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 12 0843

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
Y	WILKINSON J: "UNDERSTANDING MPEG CONCATENATION AND TRANSCODING" ABU TECHNICAL REVIEW, ASIAN BROADCASTING UNION, KUALA LUMPUR, MW, no. 177, July 1998 (1998-07), pages 3-9, XP000992282 ISSN: 0126-6209 * section "TRANSCODING" * ---	1-3, 12-14, 17-19, 27,34, 36-38, 47-49, 52-54, 62,69	H04N7/26 H04N7/50 H04N7/58 H04N7/46
Y	LIANG-WEI LEE ET AL: "ON THE ERROR DISTRIBUTION AND SCENE CHANGE FOR THE BIT RATE CONTROLOF MPEG" IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, IEEE INC. NEW YORK, US, vol. 39, no. 3, 1 August 1993 (1993-08-01), pages 545-554, XP000396331 ISSN: 0098-3063 * sections II., III. * ---	1-3, 12-14, 17-19, 27,34, 36-38, 47-49, 52-54, 62,69	
A	WO 98 44737 A (KONINKL PHILIPS ELECTRONICS NV ;PHILIPS NORDEN AB (SE)) 8 October 1998 (1998-10-08) * page 2, line 32 - page 4, line 25 * * page 14, line 21 - page 18, line 6 * ---	1-76	TECHNICAL FIELDS SEARCHED (Int.Cl.7) H04N
A	WO 95 29559 A (THOMSON CONSUMER ELECTRONICS ;OZKAN MEHMET KEMAL (US); BEYERS BILL) 2 November 1995 (1995-11-02) * page 10, line 35 - page 11, line 14 * * page 16, line 17 - page 17, line 27 * --- -/--	1-76	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	13 October 2003	Fassnacht, C

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 12 0843

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	EP 0 700 214 A (HUGHES AIRCRAFT CO) 6 March 1996 (1996-03-06) * column 8, line 8 - column 8, line 31 *	1-3, 12-14, 17-19, 27, 34-38, 47-49, 52-54, 62,69	
A	PEREIRA M ET AL: "Re-codable video" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP) AUSTIN, NOV. 13 - 16, 1994, LOS ALAMITOS, IEEE COMP. SOC. PRESS, US, vol. 3 CONF. 1, 13 November 1994 (1994-11-13), pages 952-956, XP010146085 ISBN: 0-8186-6952-7 * section "2.3.3. Improved rate-control" *	1-3, 12-14, 17-19, 27, 34-38, 47-49, 52-54, 62,69	
P,X	LIMIN WANG ET AL: "Adaptive rate control for MPEG transcoder" IMAGE PROCESSING, 1999. ICIP 99. PROCEEDINGS. 1999 INTERNATIONAL CONFERENCE ON KOBE, JAPAN 24-28 OCT. 1999, PISCATAWAY, NJ, USA,IEEE, US, 24 October 1999 (1999-10-24), pages 266-270, XP010368706 ISBN: 0-7803-5467-2 * the whole document *	1-76	TECHNICAL FIELDS SEARCHED (Int.Cl.7)
E	WO 01 84905 A (GEN INSTRUMENT CORP ;WANG LIMIN (US)) 15 November 2001 (2001-11-15) * the whole document *	1-76	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	13 October 2003	Fassnacht, C

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 12 0843

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

13-10-2003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9844737 A	08-10-1998	US 5982436 A	09-11-1999
		EP 0914745 A1	12-05-1999
		WO 9844737 A1	08-10-1998
		JP 2000511740 T	05-09-2000
		US 6208691 B1	27-03-2001
WO 9529559 A	02-11-1995	WO 9529559 A1	02-11-1995
		AU 6816094 A	16-11-1995
		EP 0803163 A1	29-10-1997
		JP 9512396 T	09-12-1997
		US 6055270 A	25-04-2000
EP 0700214 A	06-03-1996	US 5708473 A	13-01-1998
		EP 0700214 A2	06-03-1996
		JP 3072035 B2	31-07-2000
		JP 8205180 A	09-08-1996
		NO 953369 A	01-03-1996
WO 0184905 A	15-11-2001	AU 6111901 A	20-11-2001
		CA 2407682 A1	15-11-2001
		CN 1433641 T	30-07-2003
		EP 1295462 A1	26-03-2003
		TW 522740 B	01-03-2003
		WO 0184905 A2	15-11-2001

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 005 232 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
31.05.2000 Bulletin 2000/22

(51) Int. Cl.[7]: **H04N 7/26**, H04N 7/50, H04N 7/58

(21) Application number: **99120843.0**

(22) Date of filing: **25.10.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **24.11.1998 US 198867**

(71) Applicant: **General Instrument Corporation**
Horsham, Pennsylvania 19044 (US)

(72) Inventors:
, Wang, Limin
San Diego, California 92128 (US)
, Luthra, Ajay
San Diego, California 92130 (US)
, Eifrig, Robert O.
San Diego, California 92126 (US)

(74) Representative:
Hoeger, Stellrecht & Partner
Uhlandstrasse 14 c
70182 Stuttgart (DE)

(54) Rate control for an MPEG transcoder without a priori knowledge of picture type

(57) A rate control system suitable for use with a digital video transcoder, such as one conforming to the MPEG standard. The proposed rate control system starts coding with any reasonable set of assumed Group of Pictures (GOP) parameters, thereby avoiding a processing delay of about one GOP which would otherwise be incurred to extract the complete GOP structure information from a pre-compressed bit stream. In addition, the system avoids the need to store the data corresponding to the GOP, thereby reducing the memory required for transcoding. Encoding of a first picture in a sequence or GOP begins without *a priori* knowledge of the picture type of subsequent pictures. A reasonable set of GOP parameters is assumed to determine an encoding bit budget. The bit budget is gradually corrected as successive pictures are coded according to their picture types. Changes in the GOP structure of pre-compressed bitstreams can be addressed, for example, when switching channels, inserting commercials, and the like. Target rates with incorrect starting GOP parameters will converge within a few GOPs.

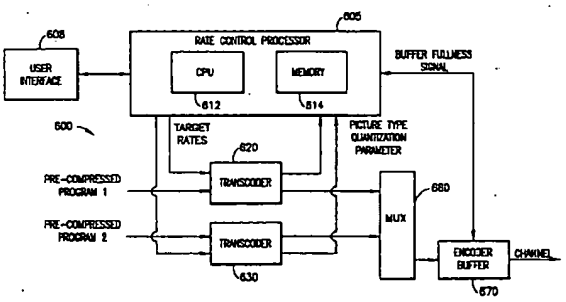


FIG.6

EP 1 005 232 A2

Description

BACKGROUND OF THE INVENTION

[0001] The present invention relates to a digital video transcoder and method for allocating bits for encoding successive pictures in a group of pictures (GOP) without *a priori* knowledge of the picture types in the GOP.

[0002] With digital video coding standards such as MPEG, input pictures can be coded in three different picture types, namely I, P and B. The three pictures require quite different numbers of bits for encoding because of the different nature of their temporal processing. Hence, an intelligent bit allocation strategy should assign a number of bits for encoding according to the picture's type. This implies a requirement of *a priori* knowledge of the picture types for a given bit budget. This requirement is not a problem for a standalone encoder as the encoder can determine the picture type for each input picture.

[0003] In fact, the encoder can plan ahead for the types of the input pictures. In contrast, a transcoder has no such *a priori* knowledge regarding a picture's type before actually processing the picture. This creates difficulty in allocating an appropriate number of bits for encoding pictures in a transcoder.

[0004] Accordingly, it would be desirable to have a method and apparatus for allocating bits for encoding pictures in a transcoder without *a priori* knowledge of the picture type. The system should avoid a processing delay of about one GOP which would otherwise be incurred to extract the complete GOP structure information from a pre-compressed bit stream. In addition, the system should avoid the need to store the data corresponding to the GOP, thereby reducing the memory required for transcoding.

[0005] The system should be compatible with both variable bit rate and constant bit rate bitstreams.

[0006] The system should be compatible with statistical multiplexing and remultiplexing systems.

[0007] The present invention provides an apparatus and method having the above and other advantages.

SUMMARY OF THE INVENTION

[0008] The present invention relates to a digital video transcoder and method for allocating bits for encoding successive pictures in a group of pictures (GOP) without *a priori* knowledge of the arrangement of picture types in the GOP.

1. Progressive Refresh Sequence:

[0009] A method for allocating bits for coding a progressive refresh sequence of pictures in a bitstream received at a digital video transcoder includes the steps of: (a) providing an assumed distance M' between a first picture of the bitstream and the next closest subsequent P-picture of the bitstream (in display order); (b) providing an assumed bit budget as a function of the assumed distance M'; (c) coding the first picture in accordance with the assumed bit budget; (d) determining a picture type of a second picture that immediately follows the first picture in the bitstream; (e) adjusting the assumed bit budget according to the picture type of the second picture; and (f) allocating bits for coding the second picture in accordance with the adjusted bit budget.

[0010] Thus, a bit budget for coding the pictures is initially assumed for the first picture, and the bit budget is updated as the second picture type is known. A target number of bits is allocated for coding each picture according to the bit budget. The first and second picture types generally indicate the picture distribution in the sequence.

[0011] The assumed bit budget is proportional to the assumed distance and a bit rate of the bitstream, and inversely proportional to a frame rate of the bitstream.

[0012] The pictures in the bitstream may form a progressive refresh sequence, where there is no GOP structure.

1.1 P-picture followed by B-picture:

[0013] When the first picture is a P-picture and the second picture is a B-picture, the method includes the further step of: determining an actual distance M" between the first picture and the next closest subsequent P-picture (in display order) according to the picture type of the first picture and the picture type of the second picture.

[0014] The actual distance M" is determined according to a difference between a temporal reference of the first picture, in a display order of the bitstream, and a temporal reference of the second picture, in the display order, plus one picture.

[0015] The method may include the further step of allocating bits for coding the remaining M"-1 pictures following the first picture in accordance with the adjusted bit budget.

[0016] The assumed bit budget is adjusted in the adjusting step (f) by +(M"-M')*"bit rate"/"frame rate", where "bit rate" is a bit rate of the bitstream, and "frame rate" is a frame rate of the bitstream.

1.2 P-picture followed by P-picture:

[0017] When the first and second pictures are P-pictures, the method includes the further step of: determining an actual distance M" between the first picture and the next closest subsequent P-picture (in display order)

according to the picture type of the first picture and the picture type of the second picture. The actual distance M" is determined according to a difference between a temporal reference of the second picture and a temporal reference of the first picture. The temporal references are determined in relation to a display order of the bitstream.

[0018] The assumed bit budget is adjusted in the adjusting step (f) by -(M'-1)*"bit rate"/"frame rate", where "bit rate" is a bit rate of the bitstream, and "frame rate" is a frame rate of the bitstream.

[0019] The method includes the further step of allocating bits for coding a series of M" pictures following an initial M" pictures that includes the first and second pictures in the bitstream in accordance with the adjusted bit budget. Thus, the adjustments to the bit budget in one series of pictures is carried over to the next series to allow correct coding of the next series.

2. Non-progressive Refresh Sequence:

[0020] Generally, the invention enables transcoding to begin by assuming a reasonable set of GOP parameters, including M, the distance between each I-picture and the next P-picture, and N, the GOP length. M and N are adjusted during transcoding as additional information becomes available regarding the GOP structure, e.g., the distribution of picture types in the GOP and the length of the GOP.

[0021] The assumed M can be corrected to the actual value within two frames, and the assumed N can be corrected to the actual value within one GOP. M should be verified, and adjusted if necessary, after each I- or P-picture, and N should be verified, and adjusted if necessary, after each I-picture. When M and/or N are adjusted, a bit rate R for coding the pictures, and n_I, n_P, and n_B, the remaining numbers of I, P and B-pictures in the current GOP, respectively, are also adjusted.

[0022] A method for allocating bits for coding a non-progressive refresh sequence of pictures in a bitstream received at a digital video transcoder, includes the steps of: (a) providing an assumed length N' of a particular group of pictures (GOP) of the bitstream; (b) providing an assumed bit budget as a function of the assumed length N'; providing an assumed distance M' between the first picture and the next closest subsequent P-picture of the bitstream (in display order); (c) allocating bits for coding a first picture of the particular GOP in accordance with the assumed bit budget, and N' and M'; (d) determining a picture type of a second picture that immediately follows the first picture; (e) adjusting the assumed bit budget according to the picture type of the second picture; and (f) allocating bits for coding the second picture in accordance with the adjusted bit budget.

[0023] The assumed bit budget is proportional to the assumed length N' and a bit rate of the bitstream, and inversely proportional to a frame rate of the bitstream.

[0024] M" may change within a GOP. Thus, the method may include the step of periodically verifying M" throughout the GOP. For example, M" may be calculated at every I-picture and/or P-picture that follows the first I-picture. The assumed bit budget is adjusted each time in accordance with M".

2.1 I-picture followed by I-picture:

[0025] When the first and second pictures are I-pictures, the method comprises the further step of: determining an actual distance M" between the first picture and the next closest subsequent I-picture of the bitstream according to the picture type of the second picture, and adjusting the assumed bit budget in accordance with M"

[0026] The actual distance M" is determined according to a difference between a temporal reference of the second picture, in a display order of the bitstream, and a temporal reference of the first picture, in the display order.

[0027] The adjusting step (e) comprises the step of: adjusting the assumed bit budget by -(N'-1)*"bit rate"/"frame rate", where "bit rate" is a bit rate of the bitstream, "frame rate" is a frame rate of the bitstream, and N=M=M".

[0028] The method comprises the further step of: allocating bits for coding the remaining pictures in the particular GOP following the second picture in accordance with the adjusted bit budget.

2.2 I-picture followed by P-picture:

[0029] When the first picture is an I-picture and the second picture is a P-picture, the method comprises the further steps of: determining an actual distance M" between the first picture and the next closest subsequent P-picture (in display order) according to the picture type of the second picture; and adjusting the assumed length N' of the GOP in accordance with the actual distance M" to provide an adjusted assumed length N"; wherein: the adjusting step (e) is responsive to the adjusted assumed length N'.

[0030] The assumed length N' of the particular GOP is adjusted in the adjusting step thereof by a factor, M(N'/M'), where M=M", to provide the adjusted assumed length N".

[0031] The adjusting step (e) comprises the step of: adjusting the assumed bit budget by a factor (N"- N')*"bit rate"/"frame rate", where "bit rate" is a bit rate of the bitstream, and "frame rate" is a frame rate of the bitstream.

[0032] The adjusting step (e) comprises the still further step of: adjusting the assumed bit budget by a factor, (M"-1')*"bit rate"/"frame rate", where "bit rate" is a bit rate of the bitstream, "frame rate" is a frame rate of the bitstream, and M=M".

[0033] Thus, both the GOP length and the distance M are used to adjust the bit budget, which in turn affects the target bit allocation for each picture.

[0034] The method includes the further step of allocating bits for coding the remaining pictures in the particular GOP following the second picture in accordance with the adjusted bit budget.

[0035] The method includes the further steps of determining if N">N or N"<N, where N is an actual length of the current GOP, after coding N" pictures in the current GOP. This is accomplished by observing the picture that is reached after coding N" pictures. For example, if the picture reached after coding N" pictures is a P-picture in the current GOP, then N"<N. If the picture reached after coding N" pictures is an I-picture, this signals the start of the next GOP, so N"=N. If an I-picture is coded before the N" are all coded, this indicates N">N. The already adjusted bit budget is then further adjusted based on whether N"<N or N">N, and bits are allocated for coding pictures in the bitstream following the N" coded pictures according to the further adjusted bit budget.

2.3 I-picture followed by B-picture:

[0036] When the first picture is an I-picture and the second picture is a B-picture, the method comprises the further steps of: determining an actual distance M" between the first picture and the next closest subsequent P-picture (in display order) according to the picture type of the second picture; and adjusting the assumed length N' of the GOP in accordance with the actual distance M" to provide an adjusted assumed length N"; wherein: the adjusting step (e) is responsive to the adjusted assumed length N'.

[0037] The assumed length N' of the particular GOP is adjusted in the adjusting step thereof by a factor, M(N'/M'), where M=M", to provide the adjusted assumed length N".

[0038] The adjusting step (e) comprises the step of: adjusting the assumed bit budget by a factor (N"- N')*"bit rate"/"frame rate", where "bit rate" is a bit rate of the bitstream, and "frame rate" is a frame rate of the bitstream.

[0039] The method comprises the further steps of: allocating bits for coding the remaining pictures in the particular GOP following the second picture in accordance with the adjusted bit budget.

[0040] Additionally, as discussed above, the already adjusted bit budget may be further adjusted based on whether N"<N or N">N, and bits may be allocated for coding pictures in the bitstream following the N" coded pictures according to the further adjusted bit budget.

[0041] Thus, the information learned regarding GOP length and M, the distance between I and/or P-pictures, in one GOP are used in coding the following GOPs. Target rates with incorrect starting GOP parameters will converge within a few GOPs.

[0042] Corresponding apparatus structures are also presented.

BRIEF DESCRIPTION OF THE DRAWINGS

[0043]

FIG. 1(a) illustrates a progressive refresh sequence of pictures in display order for continuing GOPs.

FIG. 1(b) illustrates the sequence of pictures of FIG. 1(a) in encoding order.

FIG. 2(a) illustrates a progressive refresh sequence of pictures in display order for an initial GOP in a bitstream followed by continuing GOPs.

FIG. 2(b) illustrates the sequence of pictures of FIG. 2(a) in encoding order.

FIG. 3 illustrates an MPEG-2 bitstream structure.

FIG. 4(a) illustrates a coding method for a progressive refresh picture sequence in accordance with the present invention.

FIG. 4(b) illustrates a coding method for a non-progressive refresh picture sequence in accordance with the present invention.

FIG. 4(c) illustrates a continuation of the coding method of FIG. 4(b) in accordance with the present invention.

FIG. 5 illustrates an estimated GOP length N" in accordance with the present invention.

FIG. 6 illustrates a rate control system in accordance with the present invention.

FIGs 7(a)-7(d) illustrate a C-language flowchart for implementing the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0044] The present invention relates to a digital video transcoder and method for allocating bits for encoding

successive pictures in a group of pictures (GOP) without *a priori* knowledge of the picture types in the GOP.

[0045] A novel rate control scheme and apparatus is presented for a digital video transcoder, such as one conforming to the MPEG standard, that requires no *a priori* (e.g., beforehand) knowledge of the picture types. Studies indicate that the picture target rates determined by the proposed rate control system with and without *a priori* knowledge of the picture types are very close.

[0046] MPEG defines three picture types in terms of temporal processing. They are the intra-frame coded picture (I-picture), forward temporal predictive coded picture (P-picture), and the bi-directional temporal predictive coded picture (B-picture). In MPEG video coding, pictures of an input video sequence are often grouped into groups of pictures (GOPs), where each GOP may contain one I-picture, a number of P-pictures, and optionally one or more B-pictures between I or P-pictures. The structure of a GOP can often be described by two variables: (1)N - the number of pictures in a GOP, or the GOP length, and (2) M - the distance between I or P-pictures. An example of a GOP with N = 15 and M = 3, in display order, is:

$$\cdots B_{m-2}B_{m-1}I_mB_{m+1}B_{m+2}P_{m+3}B_{m+4}B_{m+5}P_{m+6}\cdots P_{m+N-3}B_{m+N-2}B_{m+N-1}I_{m+N}B_{m+N+1}\cdots$$

where the subscripts are the picture's temporal references, and the consecutive numbers denote consecutively displayed pictures.

[0047] In general, I-pictures require many more bits for encoding than P- and B-pictures because I-pictures do not take advantage of temporal correlations between successive pictures (e.g., frames). B-pictures use the least numbers of bits for encoding because they have two temporal references and can therefore be temporally predicted more efficiently than P-pictures. P-pictures use a number of bits for encoding that is generally between that used for B- and I-pictures.

[0048] Clearly, the bits assigned to each picture for encoding should be based on its type, or its need. Furthermore, a GOP can have all the three types of pictures, and the organization of the three picture types within a GOP can be very flexible. Therefore, to wisely allocate a bit budget for encoding the pictures within a GOP, it is necessary to have *a prion* knowledge of the types of pictures in the GOP, or the GOP structure. This requirement is not a problem for a standalone encoder (i.e., an encoder that is not part of a transcoder) since the encoder can decide the type for each picture, and plan ahead in allocating bits based on the type and arrangement of pictures in a GOP. In other words, the information on GOP structure is available to the encoder.

[0049] In contrast to the standalone encoder, a transcoder converts a pre-compressed bitstream into another bitstream at a new rate. The inputs to a transcoder are pre-compressed bitstreams. Before processing a picture, the transcoder has no *a priori* knowledge of the picture's type, and hence no *a priori* knowledge of the GOP structures of the pre-compressed bitstreams.

[0050] Moreover, some transcoding schemes re-use the motion vector (MV) fields of the received pre-compressed bitstream for encoding the output bitstream at the new data rate. In this case, the picture types of the pre-compressed pictures are maintained. This means that the transcoder has no control of the GOP structures of the new output bitstreams.

[0051] Furthermore, with conventional schemes, it is possible for the GOP structure information to be extracted from the pre-compressed bitstreams. For example, after scanning a pre-compressed bitstream by the first two I-pictures, a transcoder should be able to determine the GOP structure. However, this requires a processing delay of about one GOP and a requirement of extra memory to hold all the bits for the GOP.

[0052] The present invention discloses a novel rate control system for a digital video transcoder, such as that conforming to the MPEG standard, with no requirement of *a priori* knowledge of the GOP structures of the pre-compressed bitstreams. A rate control scheme is disclosed that causes no processing delay and requires no extra memory. The scheme starts with any reasonable set of GOP parameters, and then gradually corrects these GOP parameters, if necessary, as successive pre-compressed pictures are received at the transcoder. The scheme compensates for changing GOP parameters used by the encoder(s).

[0053] Furthermore, we analyze the differences in target bit rates determined by the proposed rate control scheme with and without *a priori* knowledge of the GOP structure of the pre-compressed bitstream. We show that the target rates with and without *a priori* knowledge of GOP structure tend to be the same within only one or two GOPs. We demonstrate the differences in target rates in both numerical examples and real video sequences.

[0054] Below, in section 1, we will first overview the rate control for MPEG video coding. In section 2, we present the novel rate control for MPEG transcoder. In section 3, we analyze the differences in target rates determined by the proposed rate control with and without *a priori* knowledge of GOP structure. In section 4, we present the simulation and comparison results. Conclusions are discussed in section 5.

1. RATE CONTROL FOR MPEG VIDEO CODING

[0055] In MPEG video coding, pictures of all the three types (I, P or B) undergo the block Discrete Cosine Transform (DCT). The DCT coefficients are then quantized and variable length encoded. The resulting number of bits for each picture depends on the complexity of its corresponding block DCT image. I-pictures are coded without reference to any other pictures and, therefore, their DCT images usually require the most number of bits. P-pictures are first forward temporally predicted, and their temporal prediction difference images are then coded. The temporal prediction difference images usually contain much less information than the original pictures, and therefore need less bits for coding. B-pictures with two temporal references can often be better temporally predicted than P-pictures. Hence, B-pictures require the least number of bits. An intelligent bit allocation strategy should assign

the bits to a picture based on the picture's type, or the picture's complexity.

[0056] Let X_n be the complexity measure for picture n in a GOP of N pictures. In MPEG Test Model 5 (TM5), the complexity measure for a picture is defined as the product of the average quantization parameter used for the picture and the resulting number of bits. Logically, the bits allocated for a picture should be proportional to the picture's complexity measure, X_n, i.e.,

$$T_n = CX_n \qquad n = 0,1,...,N-1 \qquad (1)$$

where C is a constant. Here, we have assumed a linear model between the picture complexity measure and the number of bits required for coding. Furthermore, without any bias, a GOP of N pictures should be allocated

$$R_{GOP,N} = N \cdot \frac{bits\_rate}{frame\_rate} \text{ bits.} \qquad (2)$$

[0057] Clearly, the total number of bits assigned to all the pictures within a GOP of N pictures should be equal to $R_{GOP,N}$, i.e.,

$$R_{GOP,N} = \sum_{n=0}^{N-1} T_n = \sum_{n=0}^{N-1} CX_n = C\sum_{n=0}^{N-1} X_n \qquad (3)$$

[0058] From equations (1) and (3), we have

$$T_n = \frac{X_n}{\sum_{n=0}^{N-1} X_n} R_{GOP,N} \qquad (4)$$

[0059] Assume that all the pictures of the same type, i.e., I, P or B, have the same complexity measure, which is a reasonable assumption, at least for continuous scenes. We only need three complexity measures, one for each picture type. That is, X_I for I-pictures, X_P for P-pictures, and X_B for B-pictures. The target rate for frame n of type $t \in \{I,P,B\}$ and complexity measure $X_{n,t \in \{I,P,B\}}$ can therefore be written as:

$$T_n = \frac{X_{n,t}}{N_I X_I + N_P X_P + N_B X_B} R_{GOP,N} \qquad (5)$$

where N_I, N_P and N_B are the numbers of I, P and B-pictures, respectively, in the GOP. By eqn. (5), the pictures of the same type are assigned the same number of bits. This bit allocation strategy can be further modified by considering only the distribution of the remaining number of bits over the remaining pictures in the current GOP, i.e.,

$$T_n = \frac{X_{n,t}}{n_I X_I + n_P X_P + n_B X_B} R \qquad (6)$$

where n_I, n_P and n_B are, respectively, the remaining numbers of I, P and B-pictures in the current GOP, and R is the remaining number of bits, defined as

$$R \Leftarrow R - \sum_{n'=0}^{n-1} T_{n'} \qquad (7)$$

where $\Leftarrow$ denotes the assignment statement.

[0060] The remaining number of bits, R, needs to be updated at the beginning of each GOP as,

$$R = R + R_{GOP,N} = R + N \cdot \frac{bit\_rate}{frame\_rate} \qquad (8)$$

[0061] We now show that eqn. (6) also assigns the same number of bits to the pictures of the same type as long as the encoder can meet the target rate at each picture. For example, assume that frame n is a B-picture. Frame n can actually be either an I, P or B-picture. The encoder meets the nominal rate, T_n, for frame n. The target rate for frame $n+1$, Tn_{+1}, is therefore:

$$T_{n+1} = \frac{X_{s+1,t}}{n_I X_I + n_P X_P + (n_B - 1) X_B}(R - T_n) \quad (9a)$$

$$= \frac{X_{s+1,t}}{n_I X_I + n_P X_P + (n_B - 1) X_B}\left(R - \frac{X_B}{n_I X_I + n_P X_P + n_B X_B} R\right) \quad (9b)$$

$$= \frac{X_{s+1,t}}{n_I X_I + n_P X_P + (n_B - 1) X_B}\left(\frac{n_I X_I + n_P X_P + n_B X_B - X_B}{n_I X_I + n_P X_P + n_B X_B}\right) R \quad (9c)$$

$$= \frac{X_{s+1,t}}{n_I X_I + n_P X_P + n_B X_B} R \quad (9d)$$

[0062] As seen, the target rate for frame n+1, Tn_{+1}, depends upon only the type of frame n+1, i.e., $Xn_{+1}t$. In other words, regardless of the type (either I, P or B) of frame n, frame n+1 will be assigned the same number of bits according to its own type. This implies that equations (5) and (6) are actually the same if the pictures of the same type have the same complexity measure, and the encoder can meet the target rate at each picture. Equation (6) is, however, more practical as it addresses the changes in picture complexity measures and the difference between the target and actual rates at each picture.

[0063] In practice, the quality requirement may not be the same for the different picture types. For example, B-pictures are never used as references in the future temporal prediction. Hence, they can tolerate more distortion than I- and P-pictures. To address this concern of different quality requirements, we can further introduce a weighting factor for each picture type, $K_{n,t}$, as

$$T_n = CX_{n,t} / K_{n,t} \quad (10)$$

where

$$K_t = \begin{cases} K_I & for\ I\ picture \\ K_P & for\ P\ picture \\ K_B & for\ B\ picture \end{cases} \quad (11)$$

[0064] The bit allocation strategies (e.g., equations 5 and 6) now become:

$$T_n = \frac{X_{s,t}/K_{s,t}}{N_I X_I / K_I + N_P X_P / K_P + N_B X_B / K_B} R_{GOP,N} \quad (5b)$$

and

$$T_n = \frac{X_{s,t}/K_{s,t}}{n_I X_I / K_I + n_P X_P / K_P + n_B X_B / K_B} R \quad (6b)$$

[0065] Equation (6b) is the same one used in MPEG TM5 when $K_I = K_P = 1$ and $K_B = 1.4$.

[0066] The bit allocation strategy (e.g., equations 5 and 6) requires *a priori* knowledge of GOP structure, i.e., the numbers of I-, P- and B-pictures in a GOP, N_I,N_P,and N_B, or the remaining number of I, P and B-picture in the current GOP, n_I,n_P and n_B Given a pair of GOP structure variables, N and M, we can have $N_I = 1$, $N_P = NIM-N_I$ and $N_B = N-N_P-N_I$. FromN_I,N_P,and N_B, we can also calculate n_I,n_P and n_B during encoding. For example, if a P-picture is just encoded, the remaining number of P-pictures in the current GOP is decreased by 1, i.e., $n_P = n_P$-1.

[0067] As discussed, this information is unfortunately not available for a transcoder. Without a complete knowledge of the GOP structure of pre-compressed bitstreams, it is difficult to perform an intelligent bit allocation strategy. A transcoder can learn the GOP structure of a pre-compressed bitstream by scanning the bitstream through the first two I-pictures. This, however, may cause a processing delay by one GOP and require extra memory for holding the bits of the GOP.

2. RATE CONTROL FOR A TRANSCODER

[0068] We now describe a novel rate control system for a transcoder with no requirement of *a priori* knowledge of GOP structure of the pre-compressed bitstreams. The proposed rate control system causes no processing delay and requires no extra memory. It can start with any reasonable set of GOP parameters, and then gradually corrects these GOP parameters as encoding progresses, if necessary.

[0069] Table 1(a) shows all the possible picture organizations where the subscripts are the pictures' temporal references, in display order. Each picture refers to a frame or field, or portion thereof. For example, the present invention is compatible with coding of video object planes (VOPs) as known from the MPEG-4 standard. The present invention is compatible with both progressive scan and interlaced scan video.

[0070] For the sequence GOP(N=M=1), there are only I-pictures. For the sequence GOP(N,M=1), there are only I- and P-pictures. For the sequence GOP(N=M), there are only I- and B-pictures. For the sequence GOP(N,M), there are I-, P- and B-pictures. For the sequence GOP(M=1), there are only P-pictures. For the sequence GOP(M), there are only P- and B-pictures.

[0071] The first four cases in Table 1(a) are non-progressive refresh sequences since they have I-pictures. The last two case (e.g., GOP(M=1) and GOP(M)) have no I-pictures, but use a progressive refresh where P-pictures contain I-slices. An I-slice is an intra-coded portion of a picture typically extending horizontally across the picture. The location of the I-slice changes in successive P-pictures to refresh a different portion of the successive P-pictures. Hence, with a progressive refresh, the GOP length N is no longer necessary in describing the picture organization. Nevertheless,M can still be used to indicate the distance between two consecutive P-pictures. The "distance" is measured in terms of number of pictures herein, although any corresponding measure, such as time, can be used. In terms of bit allocation, we can set$N = M$, $N_I = 0$ (as there is no I pictures for progressive refresh sequence), $N_P = N/M\text{-}N_I = 1$,$N_B = N\text{-}N_P\text{-}N_I = N\text{-}N_P$ in $N = M$ bit allocation equation (5) for progressive refresh.

[0072] Table 1(b) shows the corresponding forms of picture organizations in encoding order (e.g., in bitstream order) for the picture sequences of Table 1(a). If $M = 1$ (there is no B-picture in the sequence), the encoding order is actually the same as the display order. On the other hand, if $M \neq 1$ (e.g., there are M -1 B-pictures between successive I- or P-pictures), the future I- or P-pictures have to be encoded before the M -1 B-pictures.

[0073] Furthermore, for the first GOP (e.g., initial or non-continuous GOP, designated GOP_0) in a video sequence with $M \neq 1$, there may not be any B-pictures before the first I or P-picture, as shown in Table 2(a). Table 2(a) shows possible picture organizations of the first (initial) GOP after a sequence header in a video stream, in display order. The initial GOPs are designated GOP_0. For example, for GOP_0(M), the first picture is designated P_0. An initial GOP may be processed by a transcoder when a bitstream is initially acquired or re-acquired, for example.

[0074] Table 2(b) shows the first GOP of Table 2(a) in encoding order. Here, an initial GOP with $M \neq 1$ is shown, where the second I- or P-picture is encoded before any B-picture. The second picture in encoding order will be either an I or P-picture, as shown in Table 2(b).

Table 1(a)

GOP(N = M = 1):	... I_{m-1} I_m I_{m+1} I_{m+2} ...
GOP(N,M = 1):	... P_{m-1} I_m P_{m+1} P_{m+2} ... P_{m+N-1} I_{m+N} P_{m+N+1} ...
GOP(N = M):	... B_{m-1} I_m B_{m+1} B_{m+2} ... B_{m+M-1} I_{m+M} B_{m+M+1} ...
GOP(N,M):	... B_{m-1} I_m B_{m+1} B_{m+2} ... B_{m+M-1} P_{m+M} B_{m+M+1} ... P_{m+2M} ... I_{m+N} ...
GOP(M = 1):	... P_{m-1} P_m P_{m+1} P_{m+2} ...
GOP(M):	... B_{m-1} P_m B_{m+1} B_{m+2} ... B_{m+M-1} P_{m+M} B_{m+M+1} ... P_{m+2M} ...

Table 1(b)

GOP(N = M = 1):	I_m I_{m+1} I_{m+2} ...
GOP(N,M = 1):	I_m P_{m+1} P_{m+2} ... P_{m+N-1} I_{m+N} P_{m+N+1} ...
GOP(N = M):	I_m $B_{m-(M-1)}$ $B_{m-(M-1)+1}$ ... B_{m-1} I_{m+M} B_{m+1} ...
GOP(N,M):	I_m $B_{m-(M-1)}$ $B_{m-(M-1)+1}$ ... B_{m-1} P_{m+M} B_{m+1} ... B_{m+M+1} P_{m+2M} ... I_{m+N} ...
GOP(M = 1):	P_m P_{m+1} P_{m+2} ...
GOP(M):	P_m $B_{m-(M-1)}$ $B_{m-(M-1)+1}$ ... B_{m-1} P_{m+M} B_{m+1} ... B_{m+M+1} P_{m+2M} ...

Table 2(a)

GOP_0(N = M):	I_0 B_1 B_2 ... B_{M-1} I_M B_{M+1} ... I_{2M} ...
GOP_0(N,M):	I_0 B_1 B_2 ... B_{M-1} P_M B_{M+1} ... B_{2M-1} P_{2M} ... P_{3M} ... I_N ...
GOP_0(M):	P_0 B_1 B_2 ... B_{M-1} P_M B_{M+1} ... P_{2M} ...

Table 2(b)

GOP_0(N = M):	I_0 I_M B_1 B_2 ... B_{M-1} I_{2M} B_{M+1} ...
GOP_0(N,M):	I_0 P_M B_1 B_2 ... B_{M-1} P_{2M} B_{M+1} ... B_{2M-1} P_{3M} ... I_N ...
GOP_0(M):	P_0 P_M B_1 B_2 ... B_{M-1} P_{2M} B_{M+1} ...

[0075] FIG. 1(a) illustrates a progressive refresh sequence of pictures in display order for continuing GOPs. Specifically, the progressive refresh sequence GOP(M) from Table 1(a) is illustrated for a distance between P-pictures of M=3, and for a continuing (e.g., non-initial, or open) GOP. Additionally, the GOP length is N=M=3 pictures.

[0076] In the video sequence 100, a first GOP 110 includes pictures P_7, B_8 and B_9, while a second GOP 120 includes pictures P_{10}, B_{11} and B_{12}, and a third GOP 130 includes pictures P_{13}, B_{14}, and B_{15}. The subscript "7" for P_7 is chosen simply as an example. A subsequent GOP includes P_{16} as its first picture.

[0077] FIG. 1(b) illustrates the sequence of pictures of FIG. 1(a) in encoding order. The encoded video sequence

150 corresponds to the sequence GOP(M) from Table 1(b). Since B_8 and B_9 are temporally backward predicted from P_{10}, B_8 and B_9 follow P_{10} in the encoded sequence 150. Similarly, B_{11} and B_{12} follow P_{13}, and B_{14} and B_{15} follow P_{16}.

[0078] FIG. 2(a) illustrates a progressive refresh sequence of pictures in display order for an initial GOP in a bitstream followed by continuing GOPs. Specifically, the progressive refresh sequence $GOP_0(M)$ from Table 2(a) is illustrated for a distance between P-pictures of M=3, and a GOP length is N=M=3 pictures.

[0079] In the video sequence 200, a first, initial GOP 210 includes pictures P_0, B_1 and B_2, while a second, continuing GOP 220 includes pictures P_3, B_4 and B_5, and a third, continuing GOP 230 includes pictures P_6, B_7, and B_8. A subsequent GOP includes P_9 as its first picture.

[0080] FIG. 2(b) illustrates the sequence of pictures of FIG. 2(a) in encoding order. The encoded video sequence 250 corresponds to the sequence $GOP_0(M)$ from Table 2(b). Since B_1 and B_2 are temporally backward predicted from P_3, B_1 and B_2 follow P_3 in the encoded sequence 250. Essentially, the B-pictures that would use P_0 for backward temporal prediction are not present, so P_3, B_1 and B_2 follow P_0. Similarly, B_4 and B_5 follow P_6, and B_7 and B_8 follow P_9.

[0081] FIG. 3 illustrates an MPEG-2 bitstream structure. A sequence layer 300 includes a sequence header 302, followed by a number of GOPs, e.g., GOPs 304, 306 and 308, followed by additional sequence headers and GOPs, including sequence header 312. A GOP layer 320 includes a GOP header 322, followed by data from a number of pictures, e.g., pictures 324, 326 and 328. A picture layer 330 includes a picture header 332, followed by a number of slices, e.g., slices 334 and 336. A slice layer 335 includes a slice header 337, followed by a number of macroblocks, e.g., macroblocks 338 and 339. A macroblock layer 340 includes a macroblock header 342, followed by a number of blocks, e.g., blocks 344, 346 and 348. A block layer includes DCT coefficients 352.

[0082] For a given pre-compressed bitstream, the transcoder can start processing anywhere in the bitstream. Initially, the transcoder will skip all the bits until reaching the sequence header 300, which contains information about picture size, frame rate, and the like. Optionally, the GOP header 322 may follow the sequence header 302 (for non-progressive refresh only). If a GOP header does follow the sequence header 302, the bitstream is definitely a non-progressive refresh sequence. Otherwise, the bitstream may, or may not, be of progressive refresh. The picture layer 330 then follows.

[0083] The picture immediately following the sequence header (i.e., the first picture in the GOP, i.e., picture 324) can be either an I or P-picture. If it is an I-picture, the bitstream is a non-progressive refresh sequence. Otherwise, the bitstream is a progressive refresh sequence. The bit allocation procedures of the present invention for bitstreams with progressive refresh and non-progressive refresh are slightly different. In the following two subsections, we detail both.

[0084] The following notation will be used:

N,M - the actual values (i.e., reflecting the true GOP structure) of the GOP parameters, N and M;

N',M' - the initial assumed values of the GOP parameters, N and M;

N'',M'' - the (current) updated values of the GOP parameters, N and M;

N_I,N_P,N_B - the numbers of I, P and B pictures, respectively, in a GOP;

n_I,n_P,n_B - the number of I, P and B pictures, respectively, remaining in the current GOP;

R - the number of remaining bits for the current GOP;

bit_rate - the effective output bit rate in bits per second that may be variable (we assume that bit_rate is available to transcoder);

frame_rate - the *effective* frame_rate in frames per second, which may not be the same as the frame_rate embedded in the sequence header due to the presence of the repeat_first_field (we assume that frame_rate is available to transcoder);

div - Integer division with truncation of the result toward zero; and

⇐ - Assignment statement (from the right to the left).

(i) Progressive Refresh

[0085] For a progressive refresh sequence, there will be no GOP header, and the first picture immediately following the sequence header (in encoding order) has to be a P-picture. See sequences GOP(M=1) and GOP(M) in Table 1(b) and $GOP_0(M)$ in Table 2(b). Instead of waiting to correctly determine M, we proceed to process the first P-picture with any reasonable assumed M, say M', and a bit budget of $R = R_{GOP,M} = M' \cdot (bit\_rate/frame\_rate)$ bits in calculating the target bit rate (see eqn. 6). Note that with M', we have $N_P = 1$ and $N_B = M' - N_P$. For example, refer to FIG. 1(a) or 2(a), where the distance between P-pictures is M'=3, the number of P-pictures in a GOP is one, and the

number of B-pictures in a GOP is two.

[0086] The second picture in the GOP, in encoding order, can be either a B-picture ($GOP(M)$) or a P-picture ($GOP(M = 1)$ or $GOP_0(M)$), as shown in Tables 1(b) and 2(b), respectively.

[0087] In either case, we can calculate the actual value of M from the temporal references of the first two pictures of a GOP in encoding order. Specifically, let *temp_ref_picture*1 and *temp_ref_picture*2 be the temporal references for the first and second pictures, respectively. The temporal reference may be embedded in the picture header. If the second picture is a B-picture, e.g., sequence ($GOP(M)$) in Table 1(b), the correct value of M, say M', is:

$$M' = temp\_ref\_picture\,1 - temp\_ref\_picture\,2 + 1. \qquad (12)$$

[0088] For example, in FIG. 1(b), consider P_{10} and B_8 as first and second pictures, respectively, in an encoded video sequence. Then, temp_ref_picture1 is the temporal reference of P_{10} (e.g., "10"), temp_ref_picture2 is the temporal reference of B_8 (e.g., "8"), and M"=10-8+1=3. Note that the temporal references refer to the display order of the pictures.

[0089] Otherwise, if the second picture is a P-picture, e.g., sequence ($GOP(M = 1)$ in Table 1(b), or sequence $GOP_0(M)$) in Table 2(b), the correct value of the distance between P-pictures is:

$$M' = temp\_ref\_picture\,2 - temp\_ref\_picture\,1. \qquad (13)$$

[0090] For example, in FIG. 2(b), consider P_0 and P_3 as first and second pictures, respectively, in an encoded video sequence. Then, temp_ref_picture1 is the temporal reference of P_0 (e.g., "0"), temp_ref_picture2 is the temporal reference of P_3 (e.g., "3"), and M"=3-0=3.

[0091] With a correct value of M, say M', the remaining number of bits, R (see eqn. 7), is adjusted accordingly. Specifically, if the second picture is a B-picture, as with sequence ($GOP(M)$) in Table 1(b), R is updated as

$$R = R + (M' - M) \cdot \frac{bit\_rate}{frame\_rate}. \qquad (14)$$

[0092] As seen, for $M'' > M'$ (actual distance between P-pictures is greater than assumed distance), the first P-picture has been assigned relatively more bits and, therefore, the remaining pictures will be assigned relatively fewer bits. For $M'' < M'$ (actual distance between P-pictures is less than assumed distance), the first P-picture has been assigned relatively fewer bits and the remaining pictures will be assigned relatively more bits.

[0093] On the other hand, if the second picture is a P-picture, e.g., as with sequences ($GOP(M = 1)$ in Table 1(b), or sequence $GOP_0(M)$) in Table 2(b), there are no B-pictures between the first two P-pictures in the encoded video sequence. Hence, the bits allocated for the assumed M'-1 B-pictures between two P pictures (in encoding order) should be subtracted from R, as

$$R = R - (M' - 1) \cdot \frac{bit\_rate}{frame\_rate}. \qquad (15)$$

[0094] We then set $M = M'$. If the transcoder can meet the target rate at each picture, at the end of the first M pictures, we will have $R = 0$. Hence, we will have a bit budget $R = R_{GOP,M} = M \cdot (bit\_rate/frame\_rate)$ bits for next M pictures. To address any subsequent change in M, we check the value of M after each P-picture.

[0095] FIG. 4(a) illustrates a coding method for a progressive refresh picture sequence in accordance with the present invention. The aforementioned bit allocation technique for progressive refresh sequences is summarized as follows.

[0096] At block 400, the pre-compressed digital video bitstream is received at a transcoder. The coded pictures are in an encoding order. At block 405, the bitstream is processed to detect a sequence header. At block 410, the picture type of the first picture following the sequence header is determined. If it is an I-picture, processing continues at block "A" in FIG. 4(b).

[0097] If the first picture following the sequence header is a P-picture, processing continues at block 415, where an assumed distance M' is set. M' is defined herein as a distance between the first picture following the sequence header, and the next P-picture in the display order. At block 420, an assumed bit budget R is determined based on M', the bit rate and the frame rate of the bitstream. At block 422, a target bit rate T is set in accordance with eqn. 6b. At block 425, the first picture is coded according to T. At block 430, if the next picture (e.g., the second picture) is a P-picture, the actual distance M" is determined at block 435 according to eqn. 13. At block 440, the bit budget R is adjusted using eqn. 15. At block 445, M is set to M".

[0098] At block 430, if the next picture in the sequence, e.g., the second picture, is a B-picture, the actual distance M" is determined at block 450 according to eqn. 12. At block 455, the bit budget R is adjusted using eqn. 14. At block 460, M is set to M". At block 462, the target bit rate T is set (e.g., using eqn. 6). Next, the remaining M-1 pictures (including the second picture) are coded according to T at block 465.

[0099] At block 470, the next picture is processed. This may be the first picture of the next series of M" pictures, for example. At block 475, processing returns, e.g., to block 415. However, note that, instead of blindly assuming the length and bit budgets at blocks 415 and 420, respectively, the values determined in the previous series of M" pictures may be used. Nevertheless, the actual distance M" may be verified, e.g., after every P-picture.

(ii) Non-Progressive Refresh

[0100] For non-progressive refresh, there may be an optional GOP header following the sequence header. Then, a picture layer follows. The first picture immediately following a sequence header, or a GOP header, has to be an I-picture. For an intelligent bit allocation, we need to know both N and M. We can, however, start transcoding of a pre-compressed bitstream with any reasonable pair of N and M, say N' and M', and with a bit budget of $R = R_{GOP,N} = N' \cdot (bit\_rate/frame\_rate)$ bits in determining the target rate for the first I-picture (see eqn. 6).

[0101] Note that with N' and M', we have $N_I = 1, N_P = N'/M' - N_I$ and $N_B = N' - N_I - N_P$. The actual value of M, say M'', can then be calculated from the temporal references of the first two pictures, similar to progressive refresh. If the second picture is an I- or P-picture, eqn. (13) is used in calculating the value of M. Otherwise, the second picture is a B-picture, and eqn. (12) is used. We then set $M = M''$.

[0102] If the second picture is also an I-picture ($GOP(N = M = 1)$ and $GOP_0(N = M)$), we actually have had the correct N value too, i.e., $N = M$, as shown in Table 1(b) and 2(b). The remaining number of bits, R (see eqn. 7), is adjusted accordingly, as:

$$R = R - (N' - 1) \cdot \frac{bit\_rate}{frame\_rate}. \qquad (16)$$

[0103] This is because there are no P- and B-pictures between the first and second I-pictures. Hence, the bits allocated to the assumed N-1 (P and B) pictures between two I pictures should be deducted from R.

[0104] If the second picture is not an I-picture, it will take somewhat longer to obtain the correct value of N, by about one GOP. The assumed N' may not be dividable by the correct M. Hence, we need to adjust N', and accordingly, n_P and nB. We adjust N to N'' with the correct M value, and n_P and n_B, as

$$N'' = (N'/M) \cdot M; \qquad (17)$$
$$n_p = N''/M - 1; \text{ and} \qquad (18a)$$
$$n_B = N''/M - n_p - 1. \qquad (18b)$$

[0105] We then adjust the remaining number of bits, R (see eqn. 7), with N'' as

$$R = R + (N'' - N') \cdot \frac{bit\_rate}{frame\_rate}. \qquad (19)$$

[0106] Furthermore, if the second picture is a P-picture, GOP(N,M=1) or GOP_0(N,M) as shown in Tables 1(b) and 2(b), the remaining number of bits, R, needs additional adjustment as

$$R = R - (M - 1) \cdot \frac{bit\_rate}{frame\_rate} \qquad (19b)$$

[0107] This is because there are M-1 fewer B-pictures than assumed before the first I-picture.

[0108] There can be only three scenarios, i.e.,

1. N''=N, that is, the current GOP length N'' is equal to the actual GOP length N;
2. N''<N, that is, the current GOP length N'' is smaller than the actual GOP length N; and
3. N''>N, that is, the current GOP length N'' is greater than the actual GOP length N.

[0109] Clearly, if $N = N''$, we will reach an I-picture after processing N'' pictures - the beginning of the second GOP. We have a correct N, so we maintain it.

[0110] If $N'' < N$, the GOP length is shorter than the actual GOP length. The assumed GOP length is at least M pictures shorter than the actual GOP length because of the nature of the GOP structure (see Tables 1(b) and 2(b)). Here, we have assumed that the value of M will not change within a GOP.

[0111] Note that M may be checked periodically within a GOP, such as after each I- and/or P-picture, if there is a chance that M may change within a GOP. Additionally, if M changes within the GOP, N and R should be adjusted accordingly.

[0112] We extend the GOP of N'' pictures by M additional pictures, i.e.,

$$N'' = N'' + M, \qquad (21)$$

and adjust R by adding $R_{GOP,M}$ additional bits for the M additional pictures, i.e.,

$$R = R + R_{GOP,M} = R + M \cdot \frac{bit\_rate}{frame\_rate}. \qquad (22)$$

[0113] If the extended GOP length is still shorter than the actual GOP length, we repeat the above procedure until reaching the end of the actual GOP.

[0114] If $N'>N$, the GOP length is longer than the actual GOP length. At the end of the actual GOP, we will have a correct GOP length, N, and we will use the correct length N for the rest of the GOPs. Since the assumed GOP length (N') is longer than the actual GOP length (N), there will be some bits left over from the first actual GOP for coding pictures in the second actual GOP. Hence, the bits allocated for the second actual GOP need to be adjusted as:

$$R = 2R_{GOP,N'}-(R_{GOP,N'}-R)=2N\cdot\frac{bit\_rate}{frame\_rate}-(N'\cdot\frac{bit\_rate}{frame\_rate}-R) \qquad (23)$$

where $R_{GOP,N'}-R$ is the number of bits used for encoding the first N pictures.

[0115] FIG. 4(b) illustrates a coding method for a non-progressive refresh picture sequence in accordance with the present invention. The aforementioned bit allocation technique for non-progressive refresh sequences is summarized as follows.

[0116] At block "A" processing continues from block "A" of FIG. 4(a). At block 500, an assumed GOP length N' is provided, and at block 502, an assumed distance M' is provided. M' here is a distance between the first I-picture in a GOP and the next I- or P-picture in the display order. This distance may or may not be the same as the GOP length. Generally, compatible values of N' and M' should be selected, for example, N'>M' and N' is dividable by M'. At block 504, an assumed bit budget R is provided according to N', and at block 505, a target bit rate T is determined, e.g., using eqn. 6, for coding the pictures in the GOP.

[0117] At block 506, the first picture in the GOP is coded. At block 508, if the next picture, e.g., the second picture of the GOP, is an I-picture, processing continues at block 510, where the actual distance M" is determined using eqn. 13. At block 514, M is set to M", and at block 516 N is set to M since the distance M" and the GOP length are the same.

[0118] At block 518, the bit budget is adjusted according to eqn. 16, and at block 524 processing returns, e.g., to block 500. Note that, at blocks 500 and 502, N' and M' can be set to the values N and M, respectively, of the previous GOP. N and M can be verified for each GOP.

[0119] At block 508, if the next picture, e.g., the second picture in the GOP, is a P-picture, processing continues at block 530, where the actual distance M" is determined using eqn. 13. At block 532, M is set to M". At block 534, the GOP length is updated as N"=M[N'/M']. At block 536, the remaining number of B-pictures in the current GOP, n_B, is adjusted using eqn. 18. At block 538, the bit budget is adjusted using eqn. 19, and at block 540, the bit budget is adjusted again, using eqn. 19b, and processing continues at "B".

[0120] At block 508, if the next picture, e.g., the second picture in the GOP, is a B-picture, processing continues at block 542, where the actual distance M" is determined using eqn. 12. At block 544, M is set to M". At block 546, the GOP length is updated as N"=M[N'/M']. At block 548, the remaining number of B-pictures in the current GOP, n_B, is adjusted using eqn. 18. At block 550, the bit budget is adjusted using eqn. 19, and processing continues at "B".

[0121] FIG. 4(c) illustrates a continuation of the coding method of FIG. 4(b) in accordance with the present invention. Processing continues at "B" from FIG. 4(b). At block 552, the target bit rate T is set using eqn. 6, and at block 554, a counter is set to one. At block 556, the next picture in the GOP is coded. At block 558, counter is incremented. "Counter" essentially tracks the position of the current picture in the GOP. At block 560, a determination is made as to whether the end of the current GOP has been reached. If so, a determination is made at box 562 as to whether counter=N". If so, this indicates that N" is the same as N, the actual GOP length. Essentially, the assumed GOP length N" turned out to be correct. Processing continues at block 566, where the next GOP is coded.

[0122] If the condition at block 562 is false, this indicates that N">N (e.g., the assumed GOP length N" is too large), and the bit rate is adjusted at block 564 using eqn. 23. Processing continues at block 566, where the next GOP is coded.

[0123] If the end of the current GOP has not been reached yet at block 560, a determination is made at block 570 as to whether counter=N". If not, process continues at block 556. If counter=N" at block 570, this indicates that N"<N (e.g., the assumed GOP length N" is too small). At block 572, the assumed GOP length N" is incremented by M. at block 574, the bit budget is adjusted using eqn. 22, at block 576, the next M pictures are coded, and at block 578 a determination is made as to whether the end of the current GOP has been reached. If not, processing continues at block 572, where the assume GOP length N" is again incremented by M.

[0124] When the end of the current GOP is reached, at block 578, processing continues at block 566, where the next GOP is coded.

[0125] The coding method of FIG. 4(c) can be understood further with reference to FIG. 5.

[0126] FIG. 5 illustrates an estimated GOP length N" in accordance with the present invention. A sequence of pictures is shown generally, in display order, at 500. A first GOP, GOP(N,M) 520, begins at an I-picture 502, and the next GOP 580 begins at an I-picture 510. A subsequent GOP begins at an I-picture 515.

[0127] The estimated GOP length N" 530 is equal to the actual GOP length N. However, the estimated GOP length N" 540 is less than the actual GOP length N by two M distances 542 and 544. In particular, the estimated GOP length N" 540 extends from the I-picture 502 to a P-picture 504, the M-distance 542 extends from the P-picture 504 to a P-picture 506, and the M-distance 544 extends from the P-picture 506 to the I-picture 510. The estimated GOP length N" 550 extends from the I-picture 502 to a P-picture 512, and is therefore greater than the actual GOP length N.

Generally, N" should be selected initially as a multiple of the estimated distance between I and/or P-pictures.

[0128] A C-language pseudo code for the bit-allocation procedure, both with and without progressive refresh, is shown below. The same algorithm is represented by the flowchart in FIGs 7(a)-7(d). A single bitstream can switch arbitrarily between the progressive refresh and I-picture (non-progressive) refresh modes.

```
/*
* The following is a brief description of the variables:
*
* N     Estimated open GOP length (pictures)
* M     Number of consecutive B-pictures plus 1
* N'    Initial guess for the GOP length in pictures
* M'    Initial guess for the number of consecutive B-pictures plus 1
* N"    Updated estimate of the open GOP length
* M"    Updated estimate of number of consecutive B-pictures plus 1
* R     Number of bits remaining in this GOP
* n     Number of pictures remaining in this GOP
* ni    Number of I-pictures remaining in this GOP
* np    Number of P-pictures remaining in this GOP
* nb    Number of B-pictures remaining in this GOP
* Xi    Complexity of an I-picture
* Xp    Complexity of a P-picture
* Xb    Complexity of a B-picture
* Ki    I-picture complexity divider
* Kp    P-picture complexity divider
* Kb    B-picture complexity divider
* X     Complexity temporary variable
* T     Target number of bits for the current picture type
*
* pict_type     Current picture type
* prev_pict_type        Previous picture type in coding order
* temp_ref              Current picture temporal reference
```

```
* prev_temp_ref      Previous picture temporal reference (coding order)
* prog_refresh       Non-zero if progressive refresh is in progress
* GOP_flag           Non-zero if GOP header preceded the current picture
* seq_hdr_flag       Non-zero if sequence header preceded current picture
* bit_rate()    Current average output bitrate
* frame_rate()       Average display frame rate using repeat_first_field
*
* Notes:
*
*  1) The "div" operation denotes computer integer division with
*       truncation toward 0.
*
*  2) A GOP begins with a I-picture in coding order and extends to
*       the next I-picture (exclusive) regardless of whether the GOP
*       syntax is present in the bitstream.  When progressive refresh
*       is used a GOP consist of a P-picture plus the following run of
*      consecutive B-pictures.
*/

FindSequenceHeader();
N = N';          // (N' % M') == 0; N is estimated open GOP length
M = M';
R = 0;
n = 0;
prev_pict_type = B_PICTURE;
for (;;) {
        /*
```

```
        * get_picture_header() parses the video sequence and the optional
        * GOP layer, and returns the picture type and temporal reference
        * of the next coded picture.
        */
       pict_type = get_pict_header(&temp_ref, &GOP_flag, &seq_hdr_flag);
       if (seq_hdr_flag)
              prog_refresh = (pict_type != I_PICTURE);
       switch (pict_type) {

       case I_PICTURE:
              if (prev_pict_type != B_PICTURE) {
                     /*
                      * GOP(N=M=1), GOP(N,M=1) or GOP0(N=M).  Recalculate
M,
                      * start new GOP and correct R for missing B pictures.
                      */
                     M" = GOP_flag ? (temp_ref + 1) :
                            ((temp_ref – prev_temp_ref + 1024) & 0x3FF);
                     R -= n * bit_rate() / frame_rate();
                     M = M";
                     N = (prev_pict_type == I_PICTURE) ? M : (N - n);
                     n = 0;
              }
              if (prog_refresh)              // End of progressive refresh
                     N = N';
              prog_refresh = 0;
              if (n <= 0) {    // GOP length is equal to expected length
```

```
                R += N * bit_rate() / frame_rate();
            } else {                    // GOP length is shorter than expected
                N -= n;
                N = (N div M) * M;
                R += (N - n) * bit_rate() / frame_rate();
            }
            n = N;
            ni = 1;
            np = N/M - 1;
            nb = N - ni - np;
            X = Xi / Ki;
            prev_temp_ref = temp_ref;
            break;

        case P_PICTURE:
            if (prev_pict_type != B_PICTURE) {
                /*
                 * GOP(M=1) or GOP0(M)                    [progressive
refresh]
                 * GOP(N,M=1) or GOP0(N,M)                [non-progressive
refresh]
                 */
                M" = (temp_ref - prev_temp_ref + 1024) & 0x3FF;
                R  -= (M - 1) * bit_rate() / frame_rate();
                nb -= M - 1;
                n  -= M - 1;
                if ((M" != M) && !prog_refresh) {
```

```
                        N" = (N div M") * M";        // open, not actual, GOP
length
                        n +=  (N" - N) - (M" - M);
                        R += ((N" - N) - (M" - M)) * bit_rate() / frame_rate();
                        np = n div M";
                        nb = n - np;
                        N  = N";
                    }
                    M = M";
                }
                if (n <= 0) {
                    /*
                     * End of GOP in progressive refresh mode or GOP is
                     * longer than expected in non-progressive refresh
mode.
                     */
                    N += M;
                    R += M * bit_rate() / frame_rate();
                    n = M;
                    ni = 0;
                    np = 1;
                    nb = M - 1;
                }
                X = Xp / Kp;
                prev_temp_ref = temp_ref;
                break;
```

```
        case B_PICTURE:
                if (prev_pict_type != B_PICTURE) {
                        /*
                         * GOP(N=M), GOP(N,M), GOP0(N,M)  [non-progressive
refresh]
                         * GOP(M)
[progressive refresh]
                         */
                        M" = (prev_temp_ref - temp_ref + 1025) & 0x3FF;
                        if (M" != M) {
                                if (prog_refresh) {
                                        nb += M" - M;
                                        R += (M" - M) * bit_rate() / frame_rate();
                                } else {
                                        N" = (N div M") * M";
                                        n +=  N" - N;
                                        R += (N" - N) * bit_rate() / frame_rate();
                                        np = (n + 1) div M" - 1;
                                        nb = n - np;
                                        N = N";
                                }
                                M = M";
                        }
                }
                X = Xb / Kb;
                break;
        }
```

```
        /*
         * Calculate the target bit rate
         */
        T = X * R / (ni*Xi/Ki + np*Xp/Kp + nb*Xb/Kb);
        switch (pict_type) {            // Decrement pic count for each picture type
                case I_PICTURE: ni--; break;
                case P_PICTURE: np--; break;
                case B_PICTURE: nb--; break;
        }
        n = n - 1;                      // Number of pictures left in GOP
        prev_pict_type = pict_type;
        /*
         * transcode_picture() does the actual transcoding which maintains
         * in original picture type (I, P or B). This function returns
         * the actual size of the transcoded picture in bits. This func-
         * tion updates the complexity (Xi, Xp, Xb) for each picture type.
         */
        R -= transcode_picture(pict_type, T);
}
```

[0129] This pseudo-code is expressed in a flowchart 700 in FIGs 7(a)-7(d). Note that ni, np and nb are the same as n_I, n_P and n_B, respectively, as used elsewhere.

[0130] Box 702 denotes searching for the video sequence header in order to obtain parameters necessary to begin transcoding. Initialization for the loop over transcoded pictures occurs here.

[0131] Box 704 is the call to "get_pict_header()" which consumes the picture header and all higher level syntax. The picture type (I, P or B picture) is returned. "GOP_flag" is true (non-zero) if a GOP header was encountered; "seq_hdr_flag" is true (non-zero) if a video sequence header occurred before the current coded picture and after the previous coded picture.

[0132] Box 706 tests "seq_hdr_flag". If true, "prog_refresh" is set to 1 (true) if the current picture is not an I-picture in box 708

[0133] Box 710 is the switch statement based on picture types. Although not shown in the code or flowchart, an MPEG-2 D-picture is treated like an I-picture in this algorithm.

[0134] If the previous picture type was not a B-picture (box 712), then a new value of M is determined from the temporal references. If the GOP is being shortened, the excess bits due to the expected GOP length are deleted from "R" and the new values of "M", "N", "n" are set (box 714). "R" is the target number of bits (not the actual number) remaining to code the rest of the GOP. If the actual GOP size is different, the difference in bits (positive or negative) remains in "R" and affects the target size for the next GOP. This forms a feedback loop to drive the average GOP size to "N*bit_rate() / frame_rate()".

[0135] The function "bit_rate()" returns the current target transcoder output bitrate which may change due to channel capacity or statistical multiplexing considerations. The function "frame_rate()" is the average coded frame rate, which is the frame rate specified in the video sequence header modified by the effects of the repeat_first_field flag in the various coded pictures. The quantity "bit_rate() / frame_rate()" denotes the desired current average coded picture size (in bits) to be output by the transcoder. It is an average because it does not consider the different MPEG picture types. Thus the target number of bits for a GOP, "N * bit_rate() / frame_rate()",

is N (the GOP length in pictures) times the average picture size.

[0136] If the transcoder is operating in progressive refresh mode, this mode is disabled because the current picture is an I-picture and N is set to the initial guess for GOP length (boxes 716 and 718).

[0137] If the condition in 720 is true, the correct value of "N" has been used. In this case, R is updated with the number of bits needed for a new GOP of length N (box 722). If the 720 condition is false, the our previous estimate of "N" is wrong (the GOP is shorter than expected) and N and R are updated accordingly (box 724).

[0138] Box 726 resets the picture type down counters ("ni", "np", "nb") according to the GOP type, the number of picture to go in the GOP (n), calculates the normalized I-picture complexity (X) and updates the previous picture type.

[0139] Box 728 is the start of the P-picture case in the switch statement. If the previous picture was not a B-picture (box 730), then a new value of M (M") is calculated from the temporal references. The difference between the two 10-bit temporal reference number will be negative a wrap around occurred; this is why 1024 is added. When no wrap around occurs, the 0x3FF, which is the hexadecimal representation of the binary number 1111111111, removes the 2^{10}à bit leaving the positive difference; otherwise the addition of 1024 make the negative difference positive and this result is not modified by the bit-wise AND operation. Since the previous picture was not a B-picture, potential corrections to "n" (number of pictures left in the GOP), "nb" (number of B-picture left in the GOP) and "R" (number of bits allocated for the remaining picture in the GOP) may be required.

[0140] If M has changed and the transcoder is not in progressive refresh mode (box 732), then corrections to "N", "n", "np", "nb" and "R" are made in box 734.

[0141] Box 736 updates M (previously calculated in M").

[0142] If the end of GOP has been reached, (n less than or equal to zero - box 738), then the GOP is actually longer than expected. In this case, the values of "N", "R", "n", "ni", "np" and "nb" are updated to reflect an increase GOP length of M pictures (one P-picture and M-1 B-pictures) in box 740.

[0143] Box 742 calculates the normalized P-picture complexity, "X", and resets "prev_temp_ref" to be the current temporal reference.

[0144] For B-pictures, where the previous picture was not a B-picture (conditional in box 744), the new value of M (in M") is computed. (see box 746) from the difference in temporal reference. This calculation provide for the possibility that the 10-bit temporal reference may have wrapped around (hence the +1024 and the mask to 10 bits).

[0145] If the value of M has changed (condition tested in box 748), then the state variables must be updated depending on "prog_refersh" (box 750). In the progressive refresh case, "nb" and "R" need to be modified (box 752). If I-picture refresh is in progress, "N", "np", "nb", "n" and "R" all need to change (box 754).

[0146] A new normalized complexity for B-picture is computed in box 758.

[0147] Following the switch statement, the target size, "T", for the next picture is calculated in box 760. This number is the intended size of the current picture when transcoded to the new bit rate. The actual size, returned by "transcode_picture()" is used to decrement R (see box 770). Following the calculation of "T", the number of picture remaining in the GOP, "n", and the number of pictures of the current type ("ni", "np" or "nb") must be decremented (see boxes 762 to 768).

3. DIFFERENCE IN TARGET RATES

[0148] The target bit rates for a picture determined with different sets of GOP parameters can be different (see equations 5 and 6). We now analyze the differences in target rates with and without a correct starting GOP structure. For sake of simplification in analysis, we assume that:

1. The transcoder can meet the target rate at each picture. This implies that the remaining number of bits at the end of a GOP is equal to zero, and that a GOP of N pictures is assigned exactly $R_{GOP,N} = N \cdot (bit\_rate/frame\_rate)$ bits.

2. Within a GOP, all the pictures of the same type have the same complexity measure.

3. We have a correct M value. In fact, we can have a correct M right after reading the header of the second picture.

[0149] Note that the above assumptions are only for simplifying the analysis, and may not be 100% true for a real transcoder. For example, a transcoder may not 100% meet the target rate at a frame, and therefore at the end of a GOP, the remaining number of bits R may not be zero bits.

[0150] It has been shown that with conditions 1 and 2, equations (5) and (6) will be identical. That is, all the pictures of the same type in a GOP are assigned the same number of bits. We use eqn. (5) in our analysis of differences in target rates. Assume that a pre-compressed bitstream is coded with a set of GOP parameters as follows:

$$N,M \rightarrow N_I,N_P,N_B$$

[0151] For comparison, we start transcoding of the pre-compressed bitstream with two different GOP structures, one with a correct GOP structure (I) and the second with an incorrect GOP structure (II) as follows:

Starting GOP(I): $N,M \rightarrow N_I,N_P,N_B$

Starting GOP(II): $N',M' \rightarrow N_I,N_P,N_B$

[0152] Let T_n and T'_n be the target rates for frame n with GOP(I) and GOP(II), respectively. The difference between these two target rates is:

$$\Delta=T_n-T'_n=(1-\frac{T'_n}{T_n})\cdot T_n \qquad (24a)$$

$$=(1-\frac{\frac{X_{n,I}}{K_{n,I}}}{N'_I\frac{X_I}{K_I}+N'_P\frac{X_P}{K_P}+N'_B\frac{X_B}{K_B}}R_{GOP,N'}\div\frac{\frac{X_{n,I}}{K_{n,I}}}{N_I\frac{X_I}{K_I}+N_P\frac{X_P}{K_P}+N_B\frac{X_B}{K_B}}R_{GOP,N})\cdot T_n \qquad (24b)$$

$$=(1-\frac{N_I\frac{X_I}{K_I}+N_P\frac{X_P}{K_P}+N_B\frac{X_B}{K_B}}{N'_I\frac{X_I}{K_I}+N'_P\frac{X_P}{K_P}+N'_B\frac{X_B}{K_B}}\cdot\frac{N'}{N})\cdot T_n \qquad (24c)$$

$$=\frac{(NN'_I-N'N_I)\frac{X_I}{K_I}+(NN'_P-N'N_P)\frac{X_P}{K_P}+(NN'_B-N'N_B)\frac{X_B}{K_B}}{(N'_I\frac{X_I}{K_I}+N'_P\frac{X_P}{K_P}+N'_B\frac{X_B}{K_B})}\cdot\frac{1}{N}\cdot T_n. \qquad (24d)$$

[0153] To better understand this difference in target bit rate, let us examine two numerical examples. In the first example, we assume the starting GOP (II) is shorter than the actual GOP, i.e.,

$$N=15\ M=3\rightarrow N_I=1\ N_P=4\ N_B=10$$
$$N'=12\ M'=3\rightarrow N'_I=1\ N'_P=3\ N'_B=8$$

[0154] If we set $K_I=K_P=1$ and $K_B=1.4$, and use the complexity measures as initialized in MPEG TM5, i.e.,

$$X_I=160\cdot bit\_rate/frame\_rate$$
$$X_P=60\cdot bit\_rate/frame\_rate \qquad (25)$$
$$X_B=40\cdot bit\_rate/frame\_rate$$

we have the difference between the two target rates as follows:

$$\Delta=T_n-T'_n=\frac{(15-12)\ x_I\ +\ (15\cdot3-12\cdot4)x_P\ +\ (15\cdot8-12\cdot10)x_B/1.4}{(x_I\ +\ 3x_P\ +\ 8x_B/1.4)}\cdot\frac{1}{15}\cdot T_n=3.5\%\cdot T \qquad (26)$$

Here, the target rate with *a prion* knowledge of the GOP structure is 3.5% higher than the rate without *a priori* knowledge of the GOP structure using the rate control system of the present invention. A shorter GOP results in smaller target rates for all three types of pictures.

[0155] In the second numerical example, the starting GOP (II) has a length longer than the actual one, i.e.,

$$N=15\ M=3\rightarrow N_I=1\ N_P=4\ N_B=10$$
$$N'=18\ M'=3\rightarrow N'_I=1\ N'_P=5\ N'_B=12$$

[0156] The difference in target rates is

$$\Delta=T_n-T'_n=\frac{(15-18)\,x_I+(15\cdot5-18\cdot4)x_P+(15\cdot12-18\cdot10)x_B/1.4}{(x_I+5x_P+12x_B/1.4)}\cdot\frac{1}{15}\cdot T_n=-1.7\% \qquad (27)$$

[0157] Here, the target rate with *a priori* knowledge of the GOP structure is 1.7% lower than the rate without *a priori* knowledge of the GOP structure using the rate control system of the present invention. A longer GOP results in the assignment of more bits to all the pictures of three types.

[0158] The differences in target bit rates will remain until the end of either the starting GOP or the actual GOP, depending on whether the starting GOP is shorter or longer than the actual GOP length.

[0159] If the starting GOP is shorter than the actual GOP (i.e. $N' < N$), at the end of the starting GOP, we will extend the GOP of N' pictures by M pictures with additional bits of $R_{GOP}M$. The remaining number of bits is therefore equal to

$$R = R+R_{GOP,M} = M\cdot\frac{bit\_rate}{frame\_rate} \qquad (28)$$

[0160] Note that we have assumed $R = 0$ at the end of the starting GOP of N' pictures. The additional M pictures actually form another GOP of $N=M$ pictures with $N'_I = 0$, $N'_P = 1$ and $N'_B = M-1$, and a bit budget of $M\cdot(bit\_rate/frame\_rate)$ bits. The differences in target rates for the M additional pictures of either P or B type (see eqn. 24) can be simplified as:

$$\Delta = T_n - T'_n = \frac{(-MN_I)\frac{X_I}{K_I}+(N-MN_P)\frac{X_P}{K_P}+(NN'_B-MN_B)\frac{X_B}{K_B}}{(\frac{X_P}{K_P}+N'_B\frac{X_B}{K_B})}\cdot\frac{1}{N}\cdot T_n \qquad (29)$$

[0161] For the first numerical example, the difference in target bit rates is:

$$\Delta=T_n-T'_n=\frac{(-3)\,x_I+(15-3\cdot4)x_P+(15\cdot2-3\cdot10)x_B/1..4}{(x_P+2x_B/1.4)}\cdot\frac{1}{15}\cdot T_n=17.0\%\cdot T_n \qquad (30)$$

[0162] Here, the target rate with *a priori* knowledge of the GOP structure is 17.0% higher than the rate without *a priori* knowledge of the GOP structure using the rate control system of the present invention. Relatively more bits are assigned to the pictures of the extended GOP since relatively fewer bits were assigned to the previous pictures. If the extended GOP of M pictures still cannot end the actual GOP of N pictures, we repeat the above procedure until reaching the end of the actual GOP. The differences in target rates for the pictures in the extended GOPs of M pictures will remain the same because all the extended GOPs of M pictures have the same pattern of picture organization and the same bit budget.

[0163] At the end of the first actual GOP of N pictures and also at the end of the last extended GOP of M pictures, we have complete information about the actual GOP structure with the remaining number of bits $R = 0$. Hence, from the second GOP, there will be no difference in target rates. Two target rates with and without *a priori* knowledge of GOP structure using the rate control system of the present invention will be the same from the second actual GOP.

[0164] On the other hand, if the assumed starting GOP length is longer than the actual GOP length (i.e. $N'>N$), at the end of the actual GOP, we have the complete information about the actual GOP structure, but the remaining number of bits $R \neq 0$. That is, there will be some left over bits to allocate over the second (actual) GOP. Hence, instead of simply giving $R_{GOP}N$ bits to the second (actual) GOP, we have to adjust R as $R = 2R_{GOP,N}-(R_{GOP,N}-R)$ (see eqn. 23). The target bit rates with correct and incorrect starting GOP structures will still not be the same in the second (actual) GOP. The difference in target rates in the second (actual) GOP is

$$\Delta = T_n-T'_n \qquad (31a)$$

$$=\frac{\frac{X_{n,I}}{K_{n,I}}}{N_I\frac{X_I}{K_I}X_I+N_P\frac{X_P}{K_P}+N_B\frac{X_B}{K_B}}(R_{GOP,N}-(2R_{GOP,N}-(R_{GOP,N'}-R))) \qquad (31b)$$

$$= \frac{\frac{X_{n,t}}{K_{n,t}}}{N_I \frac{X_I}{K_I} X_I + N_P \frac{X_P}{K_P} + N_B \frac{X_B}{K_B}} (-R_{GOP,N} + (R_{GOP,N'} - R)) \qquad (31c)$$

[0165] Here, R_{GOP} N-R is the number of bits assigned for the first N pictures, which is equal to:

$$R_{GOP,N'} - R = \frac{N_I \frac{X_I}{K_I} + N_P \frac{X_P}{K_P} + N_B \frac{X_B}{K_B}}{N'_I \frac{X_I}{K_I} + N'_P \frac{X_P}{K_P} + N'_B \frac{X_B}{K_B}} R_{GOP,N'} . \qquad (32)$$

[0166] Hence, the difference in target rate becomes:

$$\Delta = \frac{\frac{X_{n,t}}{K_{n,t}}}{N_I \frac{X_I}{K_I} + N_P \frac{X_P}{K_P} + N_B \frac{X_B}{K_B}} (-R_{GOP,N} + \frac{N_I \frac{X_I}{K_I} + N_P \frac{X_P}{K_P} + N_B \frac{X_B}{K_B}}{N'_I \frac{X_I}{K_I} + N'_P \frac{X_P}{K_P} + N'_B \frac{X_B}{K_B}} R_{GOP,N'}) \qquad (33a)$$

$$= -\frac{(NN'_I - N'N_I)\frac{X_I}{K_I} + (NN'_P - N'N_P)\frac{X_P}{K_P} + (NN'_B - N'N_B)\frac{X_B}{K_B}}{(N'_I \frac{X_I}{K_I} + N'_P \frac{X_P}{K_P} + N'_B \frac{X_B}{K_B})} \cdot \frac{1}{N} \cdot T_n \qquad (33b)$$

[0167] By comparing equations (24) and (33), we should see the differences in target rates in the first and second actual GOP have the same absolute values, but different signs. In fact, a longer starting GOP uses more bits for the first actual GOP and therefore leaves less bits for the second actual GOP. For the second numerical example, the differences in target rates for pictures in the second actual GOP are:

$$\Delta = T_n - T'_n = 1.7\% \cdot T_n \qquad (34)$$

[0168] At the end of the second actual GOP, the remaining number of bits R = 0. The third actual GOP will be assigned R_{GOP} N bits. From the third actual GOP, there will be no difference in target rates.

[0169] Target rates with respect to frame number for the two numerical examples were analyzed. The target rate with a correct starting GOP(N = 15, M = 3) was the benchmark for comparison. Comparative data was obtained for a starting GOP(N = 12, M = 3), and for a starting GOP(N = 18, M = 3). It was verified that all the pictures of the same type are assigned the same number of bits. Moreover, if the starting GOP is shorter than the actual GOP, fewer bits are assigned to the pictures before the end of the starting GOP, and more bits are assigned to the pictures of an extended GOP of M pictures. There is no difference in target rates starting from the second actual GOP.

[0170] On the other hand, if the starting GOP is longer than the actual one, slightly more bits are assigned to the pictures in the first actual GOP and fewer bits for the pictures in the second actual GOP. From the third actual GOP, the target rates are the same.

4. SIMULATION RESULTS

[0171] Simulations were conducted in evaluating the rate control system of the present invention. Test sequences were first coded at a bit rate of 15 Mbits/sec. with different sets of GOP parameters, as shown in Table I. The compressed bitstreams were then transcoded to a new rate of 3 Mbits/s with a starting GOP of N = 15 and M = 3.

	GOP1	GOP2	GOP3
N	30	12	8

	GOP1	GOP2	GOP3
M	3	2	1

[0172] Target rates were analyzed for a correct starting set of GOP parameters, and for a starting GOP of $N = 15$ and $M = 3$. Note that, in practice, a transcoder (or encoder) may not always meet the target rates at each frame. Hence, the remaining number of bits at the end of each GOP, R, may not be zero. The non-zero R will be passed to the next GOP (see eqn. 8). This implies that GOPs may be given slightly different number of bits. It was seen that a longer starting GOP assigns more bits for the pictures in the first GOP, as compared with a correct starting GOP. Hence, fewer bits are left for the pictures in the second GOP. The target rates with correct and incorrect GOP starts converging at the third GOP. On the other hand, a shorter starting GOP results in smaller target rates for the first few pictures and then more bits for the rest in the first GOP. Target rates become close from the second GOP.

[0173] To prove the effectiveness of the bit rate allocation system under a variety of test conditions, a sequence was coded with various GOP lengths, e.g., $N = 15, 12, 12, 18, 18, 15, 15, 9, 9, 30$. Target rates were analyzed at each GOP for the correct GOP length, and by using the previous GOP structure length as the starting GOP length for each GOP. The results were close with this stress test.

[0174] FIG. 6 illustrates a rate control system in accordance with the present invention. Pre-compressed bitstreams, e.g., program 1 and program 2, are provided to respective transcoders 620 and 630. One or several bitstreams and transcoders may be present. The bitstreams may be received real-time at the transcoders from a remote source, such as via a satellite broadcast, or may be provided from a local storage medium, for example.

[0175] The transcoders 620, 630 partially decompress the respective bitstreams, and encode the partially decompressed data at a different data rate, typically by using a different quantization parameter, according to a target bit rate signal provided by a rate control processor 605.

[0176] The rate control processor 605 receives information from the transcoders, including the picture type of the current picture, and the quantization parameter used to encode the current picture. This information is processed as described herein to set the target bit rates. The rate control processor 605 may include known processing circuitry, such as a central processing unit (CPU) 612, and a memory 614. The memory 614 may be a non-volatile memory that stores data, for example, for providing the initial distances M' and N' discussed previously. The rate control processor includes computing circuitry for executing the C-language like pseudo-code discussed previously.

[0177] A user interface 608 may communicate with the rate control processor 605 to set the default distances M' and N' or other relevant parameters in the rate control process.

[0178] The transcoded data from the transcoders 620, 630 is provided to a multiplexer (MUX) 660, and then to an encoder buffer 670. A buffer fullness signal is provided from the encoder buffer 670 to the rate control processor 605. The encoded data is finally transmitted over a channel at its new transcoded data rate.

5. CONCLUSION

[0179] A novel rate control system suitable for use with a digital video transcoder, such as one conforming to the MPEG standard, has been presented. The proposed rate control causes no processing delay and requires no extra memory. It can start with any reasonable set of GOP parameters, and then gradually corrects them when necessary as successive pictures are coded. Hence, it is able to address changes in the GOP structure of pre-compressed bitstreams, for example, when switching channels, inserting commercials, and the like. It has been shown that the target rates with correct and incorrect starting GOP parameters will converge within one or two GOPs. The differences in target rates are within a fairly small margin. Moreover, similar target rates correspond to a similar PSNR or picture quality.

[0180] Although the invention has been described in connection with various specific embodiments, those skilled in the art will appreciate that numerous adaptations and modifications may be made thereto without departing from the spirit and scope of the invention as set forth in the claims.

Claims

1. A method for allocating bits for coding pictures in a bitstream received at a digital video transcoder, comprising the steps of:

   (a) providing an assumed distance M' between a first picture of said bitstream and the next closest subsequent P-picture of said bitstream in a display order of said bitstream;

   (b) providing an assumed bit budget as a function of said assumed distance M';

   (c) coding said first picture in accordance with said assumed bit budget;

(d) determining a picture type of a second picture that immediately follows said first picture in said bitstream in an encoding order of said bit stream;

(e) adjusting said assumed bit budget according to said picture type of said second picture; and

(f) allocating bits for coding said second picture in accordance with said adjusted bit budget.

2. The method of claim 1, wherein:

said assumed bit budget is proportional to said assumed distance and a bit rate of said bitstream, and inversely proportional to a frame rate of said bitstream.

3. The method of claim 1 or 2, wherein:

said pictures in said bitstream form a progressive refresh sequence.

4. The method of one of claims 1 to 3, wherein said first picture is a P-picture and said second picture is a B-picture, comprising the further step of:

determining an actual distance M" between said first picture and said next closest subsequent P-picture, in said display order, according to said picture type of said second picture.

5. The method of claim 4, wherein:

said actual distance M" is determined according to a difference between a temporal reference of said second picture, in said display order, and a temporal reference of said first picture, in said display order, plus one picture.

6. The method of claim 4 or 5, comprising the further step of:

allocating bits for coding the remaining M"-1 pictures following said first picture in accordance with said adjusted bit budget.

7. The method of one of claims 4 to 6, wherein:

said adjusting step (e) comprises the step of adjusting said assumed bit budget by +(M"-M')*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

8. The method of claim 1, wherein said first and second pictures are P-pictures, comprising the further step of:

determining an actual distance M" between said first picture and said next closest subsequent P-picture, in said display order, according to said picture type of said second picture.

9. The method of claim 8, wherein:

said actual distance M" is determined according to a difference between a temporal reference of said second picture, in said display order, and a temporal reference of said first picture, in said display order.

10. The method of claim 8 or 9, wherein said adjusting step (e) comprises the step of:

adjusting said assumed bit budget by -(M'-1)*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

11. The method of one of claims 8 to 10, comprising the further step of:

allocating bits for coding the remaining M"-1 pictures following said first picture in accordance with said adjusted bit budget.

12. The method of one of claims 1 to 5 or 8 to 11, comprising the further step of:

allocating bits for coding a series of M" pictures following an initial M" picture that includes said first and second pictures in said bitstream in accordance with said adjusted bit budget.

13. A method for allocating bits for coding a sequence of pictures in a bitstream received at a digital video transcoder, comprising the steps of:

(a) providing an assumed length N' of a particular group of pictures (GOP) of said bitstream;

(b) providing an assumed bit budget as a function of said assumed length N';

(c) allocating bits for coding a first picture of said particular GOP in accordance with said assumed bit budget;

(d) determining a picture type of a second picture that immediately follows said first picture;

(e) adjusting said assumed bit budget according to said picture type of said second picture; and

(f) allocating bits for coding said second picture in accordance with said adjusted bit budget.

14. The method of claim 13, wherein:

said assumed bit budget is proportional to said assumed length N' and a bit rate of said bitstream, and inversely proportional to a frame rate of said bitstream.

15. The method of claim 13 or 14, wherein said first and second pictures are I-pictures, comprising the further steps of:

determining an actual distance M" between said first picture and the next closest subsequent I-picture of said bitstream according to the picture type of said second picture; and

adjusting said assumed bit budget in accordance with said actual distance M".

16. The method of claim 15, wherein:

said actual distance M" is determined according to a difference between a temporal reference of said second picture, in a display order of said bitstream, and a temporal reference of said first picture, in said display order.

17. The method of one of claims 13 to 16, wherein said adjusting step (e) comprises the step of:

adjusting said assumed bit budget by -(N'-1)*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

18. The method of one of claims 13 to 17, comprising the further step of:

allocating bits for coding the remaining pictures in said particular GOP following said second picture in accordance with said adjusted bit budget.

19. The method of one of claims 13 to 18, comprising the further step of:

providing an assumed distance M' between said first picture and the next closest subsequent P-picture of said bitstream, in a display order of said bitstream;
wherein said bits are allocated for coding said first picture in said step (c) in accordance with said assumed distance M'.

20. The method of claim 19, wherein said first picture is an I-picture and said second picture is a P-picture, comprising the further steps of:

determining an actual distance M" between said first picture and the next closest subsequent P-picture, in said display order, according to said picture type of said second picture; and

adjusting said assumed length N' of said GOP in accordance with said actual distance M" to provide an adjusted assumed length N"; wherein:

said adjusting step (e) is responsive to said adjusted assumed length N'.

21. The method of claim 20, wherein:

said assumed length N' of said particular GOP is adjusted in said adjusting step thereof by a factor, M(N'/M'), to provide said adjusted assumed length N", where M=M".

22. The method of claim 20 or 21, wherein said adjusting step (e) comprises the step of:

adjusting said assumed bit budget by a factor(N"-N')*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

23. The method of claim 20 or 21, wherein said adjusting step (e) comprises the further step of:

adjusting said assumed bit budget by a factor, (M"-1')*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

24. The method of one of claims 20 to 23, comprising the further step of:

allocating bits for coding the remaining pictures in said particular GOP following said second picture in accordance with said adjusted bit budget.

25. The method of one of claims 20 to 24, comprising the further steps of:

determining if N">N, an actual length of said particular GOP, after coding N" pictures in said particular GOP; and, if N">N:

further adjusting said adjusted bit budget according to N and N"; and

allocating bits for coding pictures in said bitstream following the N" coded pictures according to said further adjusted bit budget.

26. The method of one of claims 20 to 24, comprising the further steps of:

determining if N"<N, an actual length of said particular GOP, after coding N" pictures in said particular GOP; and, if N"<N:

further adjusting said adjusted bit budget according to M=M"; and

allocating bits for coding pictures in said bitstream following the N" coded pictures according to said further adjusted bit budget.

27. The method of claim 13, wherein said first picture is an I-picture and said second picture is a B-picture, comprising the further step of:

providing an assumed distance M' between said first picture and the next closest subsequent P-picture of said bitstream, in a display order of said bitstream;
wherein said bits are allocated for coding said second picture in accordance with said assumed distance M'.

28. The method of claim 27, comprising the further steps of:

determining an actual distance M" between said first picture and the next closest subsequent P-picture, in said display order, according to said picture type of said second picture; and

adjusting said assumed length N' of said GOP in accordance with an actual distance M=M" to provide an adjusted assumed length N"; wherein:

said adjusting step (e) is responsive to said adjusted assumed length N'.

29. The method of claim 28, wherein:

said assumed length N' of said particular GOP is adjusted in said adjusting step thereof by a factor, M(N'/M'), to provide said adjusted assumed length N".

30. The method of claim 28 or 29, wherein said adjusting step (e) comprises the step of:

adjusting said assumed bit budget by a factor(N"-N')*"bit rate"/'frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

31. The method of one of claims 28 to 30, comprising the further step of:

allocating bits for coding the remaining pictures in said particular GOP following said second picture in accordance with said adjusted bit budget.

32. The method of one of claims 28 to 31, comprising the further step of:

determining if N">N, an actual length of said particular GOP, after coding N" pictures in said particular GOP; and, if N">N:

further adjusting said adjusted bit budget according to N and N"; and

allocating bits for coding pictures in said bitstream following the N" coded pictures according to said further adjusted bit budget.

33. The method of one of claims 28 to 32, comprising the further steps of:

determining if N"<N, an actual length of said particular GOP, after coding N" pictures in said particular GOP; and, if N"<N:

further adjusting said adjusted bit budget according to M=M"; and

allocating bits for coding pictures in said bitstream following the N" coded pictures according to said further adjusted bit budget.

34. The method of one of claims 13 to 33, comprising the further step of:

allocating bits for coding a GOP following said particular GOP in accordance with said adjusted bit budget.

35. The method of one of claims 13 to 34, comprising the further step of:

accounting for changes in an actual distance M" between I-pictures and/or P-pictures in said particular GOP by determining an actual distance M" between an I-picture or P-picture that follows said first picture in said particular GOP, in a display order of said bitstream, and a subsequent I-picture or P-picture in said particular GOP; and

adjusting said assumed bit budget in accordance with said actual distance M".

36. An apparatus for allocating bits for coding pictures in a bitstream received at a digital video transcoder, comprising:

(a) means for providing an assumed distance M' between a first picture of said bitstream and the next closest subsequent P-picture of said bitstream in a display order of said bitstream;

(b) means for providing an assumed bit budget as a function of said assumed distance M';

(c) means for coding said first picture in accordance with said assumed bit budget;

(d) means for determining a picture type of a second picture that immediately follows said first picture in said bitstream in an encoding order of said bit stream;

(e) means for adjusting said assumed bit budget according to said picture type of said second picture; and

(f) means for allocating bits for coding said second picture in accordance with said adjusted bit budget.

37. The apparatus of claim 36, wherein:

said assumed bit budget is proportional to said assumed distance and a bit rate of said bitstream, and inversely proportional to a frame rate of said bitstream.

38. The apparatus of claim 36 or 37, wherein:

said pictures in said bitstream form a progressive refresh sequence.

39. The apparatus of one of claims 36 to 38, wherein said first picture is a P-picture and said second picture is a B-picture, further comprising:

means for determining an actual distance M" between said first picture and said next closest subsequent P-picture, in said display order, according to said picture type of said second picture.

40. The apparatus of claim 39, wherein:

said actual distance M" is determined according to a difference between a temporal reference of said second picture, in said display order, and a temporal reference of said first picture, in said display order, plus one picture.

41. The apparatus of claim 39 or 40, further comprising:

means for allocating bits for coding the remaining M"-1 pictures following said first picture in accordance with said adjusted bit budget.

42. The apparatus of one of claims 39 to 41, wherein:

said adjusting means (e) comprises means for adjusting said assumed bit budget by +(M"-M')*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

43. The apparatus of one of claims 36 to 42, wherein said first and second pictures are P-pictures, further comprising:

means for determining an actual distance M" between said first picture and said next closest subsequent P-picture, in said display order, according to said picture type of said second picture.

44. The apparatus of claim 43, wherein:

said actual distance M" is determined according to a difference between a temporal reference of said second

picture, in said display order, and a temporal reference of said first picture, in said display order.

45. The apparatus of claim 43 or 44, wherein said adjusting means (e) comprises:

means for adjusting said assumed bit budget by - (M'-1)*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

46. The apparatus of one of claims 43 to 45, further comprising:

means for allocating bits for coding the remaining M"-1 pictures following said first picture in accordance with said adjusted bit budget.

47. The apparatus of one of claims 36 to 45, further comprising:

means for allocating bits for coding a series of M" pictures following an initial M" pictures that includes said first and second pictures in said bitstream in accordance with said adjusted bit budget.

48. An apparatus for allocating bits for coding a sequence of pictures in a bitstream received at a digital video transcoder, comprising:

(a) means for providing an assumed length N' of a particular group of pictures (GOP) of said bitstream;

(b) means for providing an assumed bit budget as a function of said assumed length N';

(c) means for allocating bits for coding a first picture of said particular GOP in accordance with said assumed bit budget;

(d) means for determining a picture type of a second picture that immediately follows said first picture;

(e) means for adjusting said assumed bit budget according to said picture type of said second picture; and

(f) means for allocating bits for coding said second picture in accordance with said adjusted bit budget.

49. The apparatus of claim 48, wherein:

said assumed bit budget is proportional to said assumed length N' and a bit rate of said bitstream, and inversely proportional to a frame rate of said bitstream.

50. The apparatus of claim 48 or 49, wherein said first and second pictures are I-pictures, further comprising:

means for determining an actual distance M" between said first picture and the next closest subsequent I-picture of said bitstream according to the picture type of said second picture; and

means for adjusting said assumed bit budget in accordance with said actual distance M".

51. The apparatus of claim 50, wherein:

said actual distance M" is determined according to a difference between a temporal reference of said second picture, in a display order of said bitstream, and a temporal reference of said first picture, in said display order.

52. The apparatus of one of claims 48 to 51, wherein said adjusting means (e) comprises:

means for adjusting said assumed bit budget by - (N'-1)*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

53. The apparatus of one of claims 48 to 52, further comprising:

means for allocating bits for coding the remaining pictures in said particular GOP following said second picture in accordance with said adjusted bit budget.

54. The apparatus of one of claims 48 to 53, further comprising:

means for providing an assumed distance M' between said first picture and the next closest subsequent P-picture of said bitstream, in a display order of said bitstream;
wherein said bits are allocated for coding said first picture by said coding means (c) in accordance with said assumed distance M'.

55. The apparatus of claim 54, wherein said first picture is an I-picture and said second picture is a P-picture, further comprising:

means for determining an actual distance M" between said first picture and the next closest subsequent P-picture, in said display order, according to said picture type of said second picture; and

means for adjusting said assumed length N' of said GOP in accordance with said actual distance M" to provide an adjusted assumed length N"; wherein:

said adjusting means (e) is responsive to said adjusted assumed length N'.

56. The apparatus of claim 55, wherein:

said assumed length N' of said particular GOP is adjusted by said adjusting means thereof by a factor, M(N'/M'), to provide said adjusted assumed length N", where M=M".

57. The apparatus of claim 55 or 56, wherein said adjusting means (e) comprises:

means for adjusting said assumed bit budget by a factor (N"-N')*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

58. The apparatus of claim 57, wherein said adjusting means (e) comprises:

means for adjusting said assumed bit budget by a factor, (M"-1')*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

59. The apparatus of one of claims 55 to 58, further comprising:

means for allocating bits for coding the remaining pictures in said particular GOP following said second picture in accordance with said adjusted bit budget.

60. The apparatus of one of claims 55 to 59, further comprising:

means for determining if N">N, an actual length of said particular GOP, after coding N" pictures in said particular GOP;

means for further adjusting said adjusted bit budget according to N and N" if N">N; and

means for allocating bits for coding pictures in said bitstream following the N" coded pictures according to said further adjusted bit budget if N">N.

61. The apparatus of one of claims 55 to 59, further comprising:

means for determining if N"<N, an actual length of said particular GOP, after coding N" pictures in said particular GOP;

means for further adjusting said adjusted bit budget according to M=M" if N"<N; and

means for allocating bits for coding pictures in said bitstream following the N" coded pictures according to said further adjusted bit budget if N"<N.

62. The apparatus of claim 48, wherein said first picture is an I-picture and said second picture is a B-picture, further comprising:

means for providing an assumed distance M' between said first picture and the next closest subsequent P-picture of said bitstream, in a display order of said bitstream;
wherein said bits are allocated for coding said second picture in accordance with said assumed distance M'.

63. The apparatus of claim 62, further comprising:

means for determining an actual distance M" between said first picture and the next closest subsequent P-picture, in said display order, according to said picture type of said second picture; and

means for adjusting said assumed length N' of said GOP in accordance with an actual distance M=M" to provide an adjusted assumed length N"; wherein:

said adjusting means (e) is responsive to said adjusted assumed length N'.

64. The apparatus of claim 63, wherein:

said assumed length N' of said particular GOP is adjusted by said adjusting means thereof by a factor, M(N'/M'), to provide said adjusted assumed length N".

65. The apparatus of claim 63 or 64, wherein said adjusting means (e) comprises:

means for adjusting said assumed bit budget by a factor (N"-N')*"bit rate"/"frame rate", where "bit rate" is a bit rate of said bitstream, and "frame rate" is a frame rate of said bitstream.

66. The apparatus of one of claims 63 to 65, further comprising:

means for allocating bits for coding the remaining pictures in said particular GOP following said second picture in accordance with said adjusted bit budget.

67. The apparatus of one of claims 63 to 66, further comprising:

means for determining if N">N, an actual length of said particular GOP, after coding N" pictures in said particular GOP;

means for further adjusting said adjusted bit budget according to N and N" if N">N; and

means for allocating bits for coding pictures in said bitstream following the N" coded pictures according to said further adjusted bit budget if N">N.

68. The apparatus of one of claims 63 to 66, further comprising:

means for determining if N"<N, an actual length of said particular GOP, after coding N" pictures in said particular GOP;

means for further adjusting said adjusted bit budget according to M=M" if N"<N; and

means for allocating bits for coding pictures in said bitstream following the N" coded pictures according to said further adjusted bit budget if N"<N.

69. The apparatus of one of claims 48 to 68, further comprising:

means for allocating bits for coding a GOP following said particular GOP in accordance with said adjusted

bit budget.

70. The apparatus of one of claims 48 to 69, further comprising:

means for accounting for changes in an actual distance M" between I-pictures and/or P-pictures in said particular GOP by determining an actual distance M" between an I-picture or P-picture that follows said first picture in said particular GOP, in a display order of said bitstream, and a subsequent I-picture or P-picture in said particular GOP; and

means for adjusting said assumed bit budget in accordance with said actual distance M".

71. A method for allocating bits for coding a sequence of pictures in a bitstream received at a digital video transcoder, comprising the steps of:

(a) providing a length of groups of pictures (GOPs) of said bitstream;

(b) providing a distance between each I-picture and a next successive P-picture in said GOPs;

(c) maintaining a count of a number of remaining I-, P- and B-pictures, respectively, in each GOP as each picture in said GOPs is coded;

(d) providing an assumed bit budget in response to said steps (a), (b) and (c) for coding pictures in said GOPs;

(e) verifying, and adjusting, if necessary, said length after each I-picture in said GOPs is coded;

(f) verifying, and adjusting, if necessary, said distance after each I-picture and/or P-picture in said GOPs is coded;

(g) verifying, and adjusting, if necessary, the count of the number of remaining I-, P- and B-pictures, in response to said steps (e) and (f); and

(g) adjusting, if necessary, said assumed bit budget in response to said steps (e), (f) and (g).

72. The method of claim 71, wherein:

said distance is assumed when initially coding said GOPs and is corrected to an actual distance within a time for coding two pictures in said bitstream.

73. The method of claim 71 or 72, wherein:

said length is assumed when initially coding said GOPs and is corrected to an actual length within a time for coding one GOP in said bitstream.

74. An apparatus for allocating bits for coding a sequence of pictures in a bitstream received at a digital video transcoder, comprising:

(a) means for providing a length of groups of pictures (GOPs) of said bitstream;

(b) means for providing a distance between each I-picture and a next successive P-picture in said GOPs;

(c) means for maintaining a count of a number of remaining I-, P- and B-pictures, respectively, in each GOP as each picture in said GOPs is coded;

(d) means for providing an assumed bit budget in response to said means (a), (b) and (c) for coding pictures in said GOPs;

(e) means for verifying, and adjusting, if necessary, said length after each I-picture in said GOPs is coded;

(f) means for verifying, and adjusting, if necessary, said distance after each I-picture and/or P-picture in said GOPs is coded;

(g) means for verifying, and adjusting, if necessary, the count of the number of remaining I-, P- and B-pictures, in response to said means (e) and (f); and

(g) means for adjusting, if necessary, said assumed bit budget in response to said means (e), (f) and (g).

75. The apparatus of claim 74, wherein:

said distance is assumed when initially coding said GOPs and is corrected to an actual distance within a time for coding two pictures in said bitstream.

76. The apparatus of claim 74 or 75, wherein:

said length is assumed when initially coding said GOPs and is corrected to an actual length within a time for coding one GOP in said bitstream.

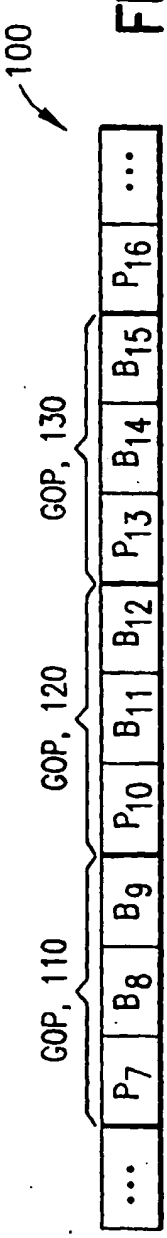


FIG.1a

FIG.1b

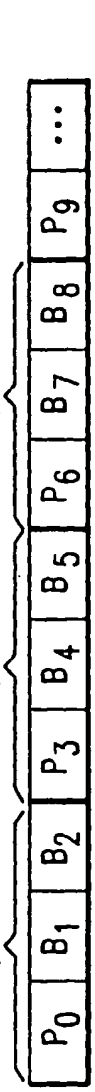


FIG.2a

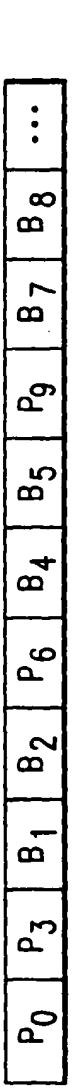


FIG.2b

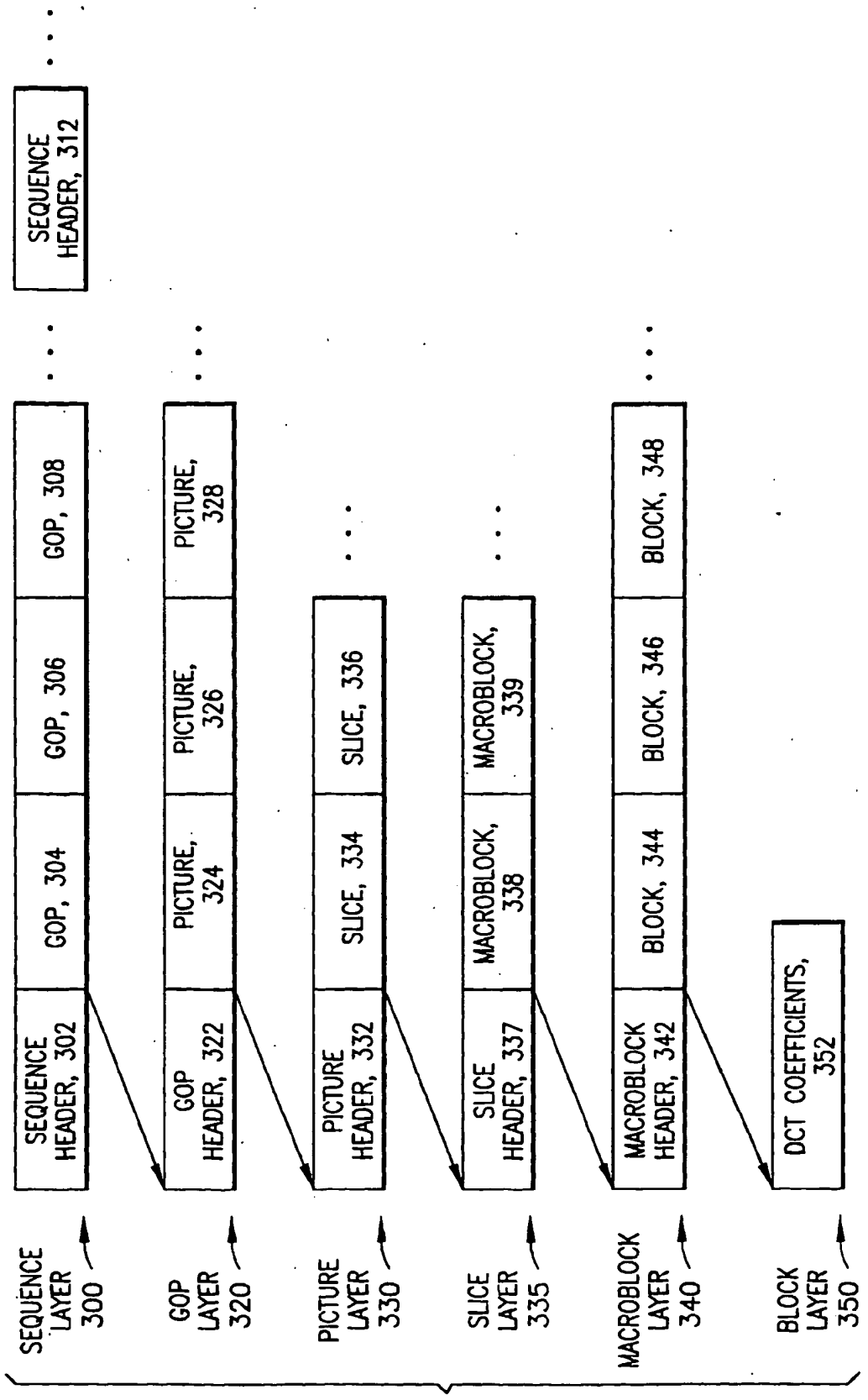

SEQUENCE LAYER 300
GOP LAYER 320
PICTURE LAYER 330
SLICE LAYER 335
MACROBLOCK LAYER 340
BLOCK LAYER 350
SEQUENCE HEADER, 302
GOP, 304
GOP, 306
GOP, 308
SEQUENCE HEADER, 312
GOP HEADER, 322
PICTURE, 324
PICTURE, 326
PICTURE, 328
PICTURE HEADER, 332
SLICE, 334
SLICE, 336
SLICE HEADER, 337
MACROBLOCK, 338
MACROBLOCK, 339
MACROBLOCK HEADER, 342
BLOCK, 344
BLOCK, 346
BLOCK, 348
DCT COEFFICIENTS, 352

FIG.3

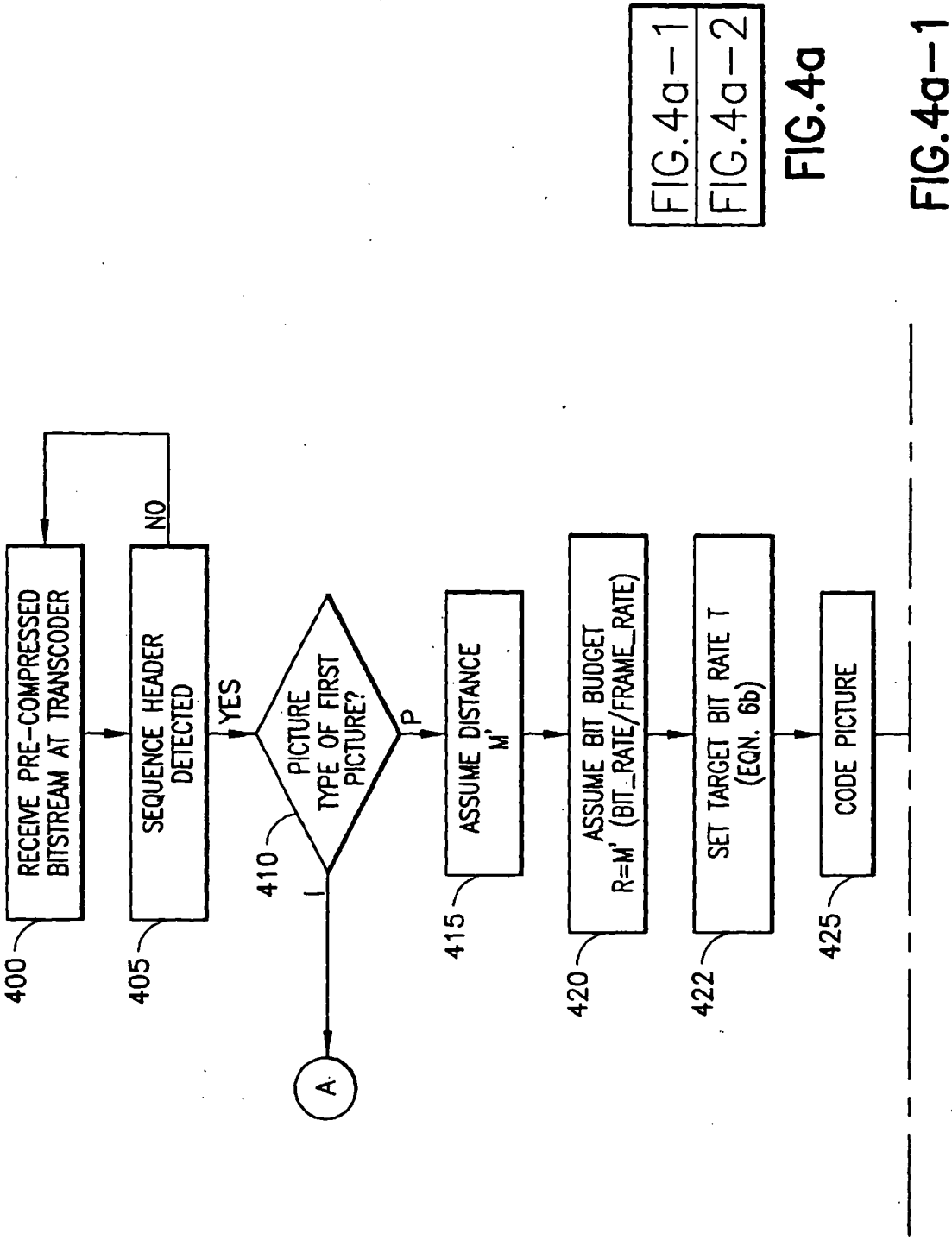


FIG.4a-1

FIG.4a

FIG.4a-1
FIG.4a-2

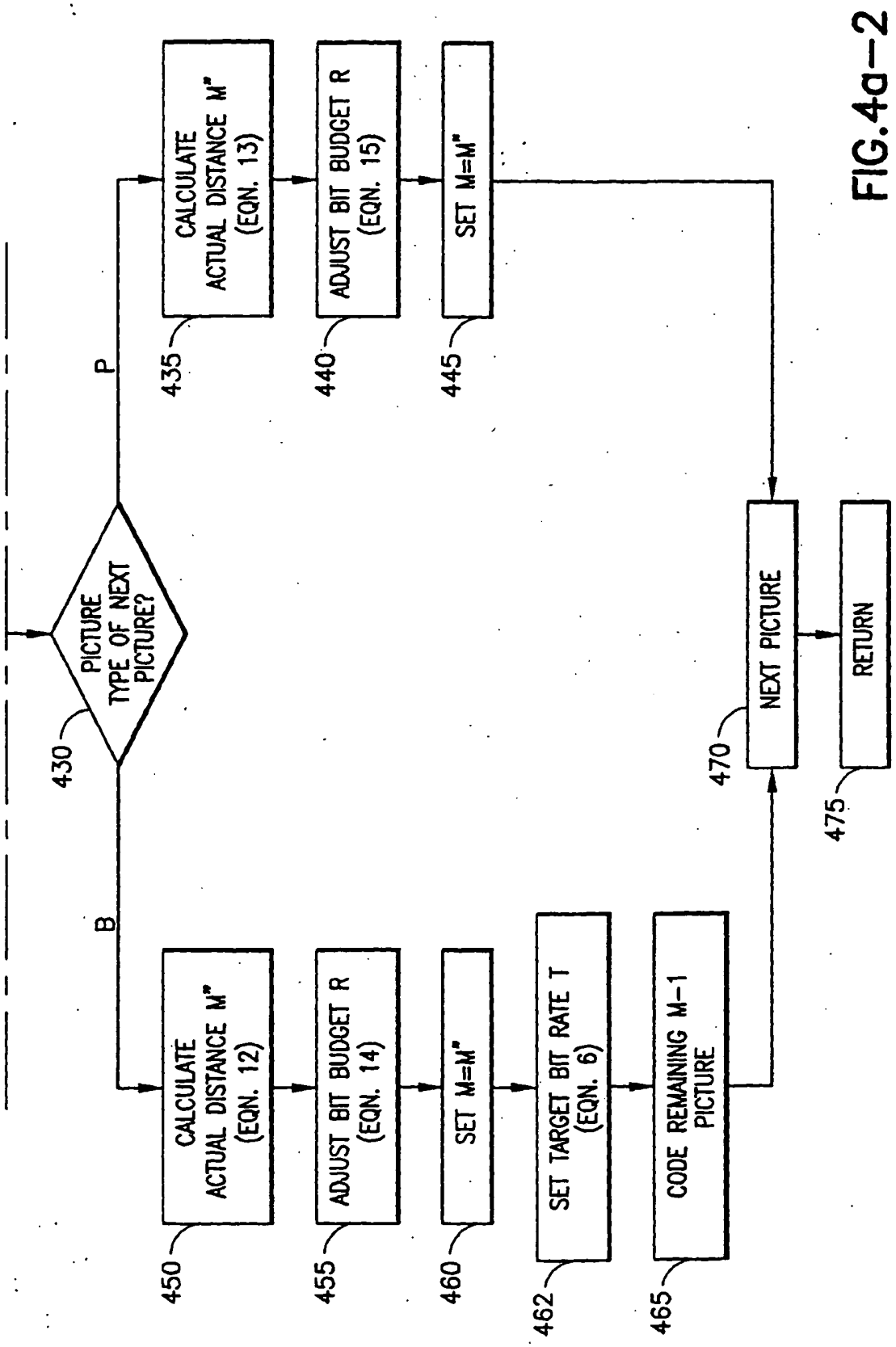

PICTURE TYPE OF NEXT PICTURE?
430
P
B
CALCULATE ACTUAL DISTANCE M" (EQN. 13)
435
ADJUST BIT BUDGET R (EQN. 15)
440
SET M=M"
445
CALCULATE ACTUAL DISTANCE M" (EQN. 12)
450
ADJUST BIT BUDGET R (EQN. 14)
455
SET M=M"
460
SET TARGET BIT RATE T (EQN. 6)
462
CODE REMAINING M-1 PICTURE
465
NEXT PICTURE
470
RETURN
475

FIG.4a-2

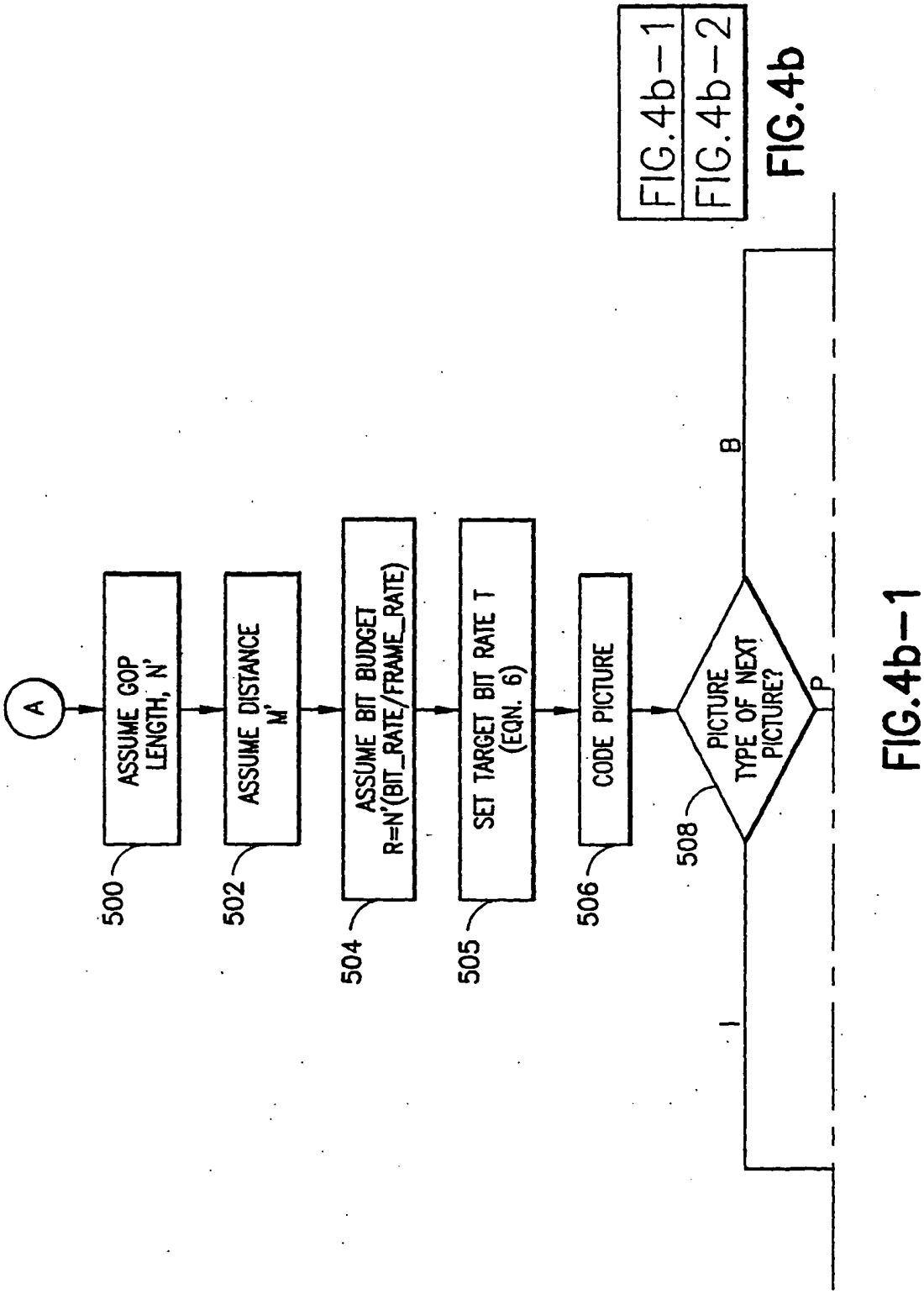


FIG.4b-1

FIG.4b-1
FIG.4b-2

FIG.4b

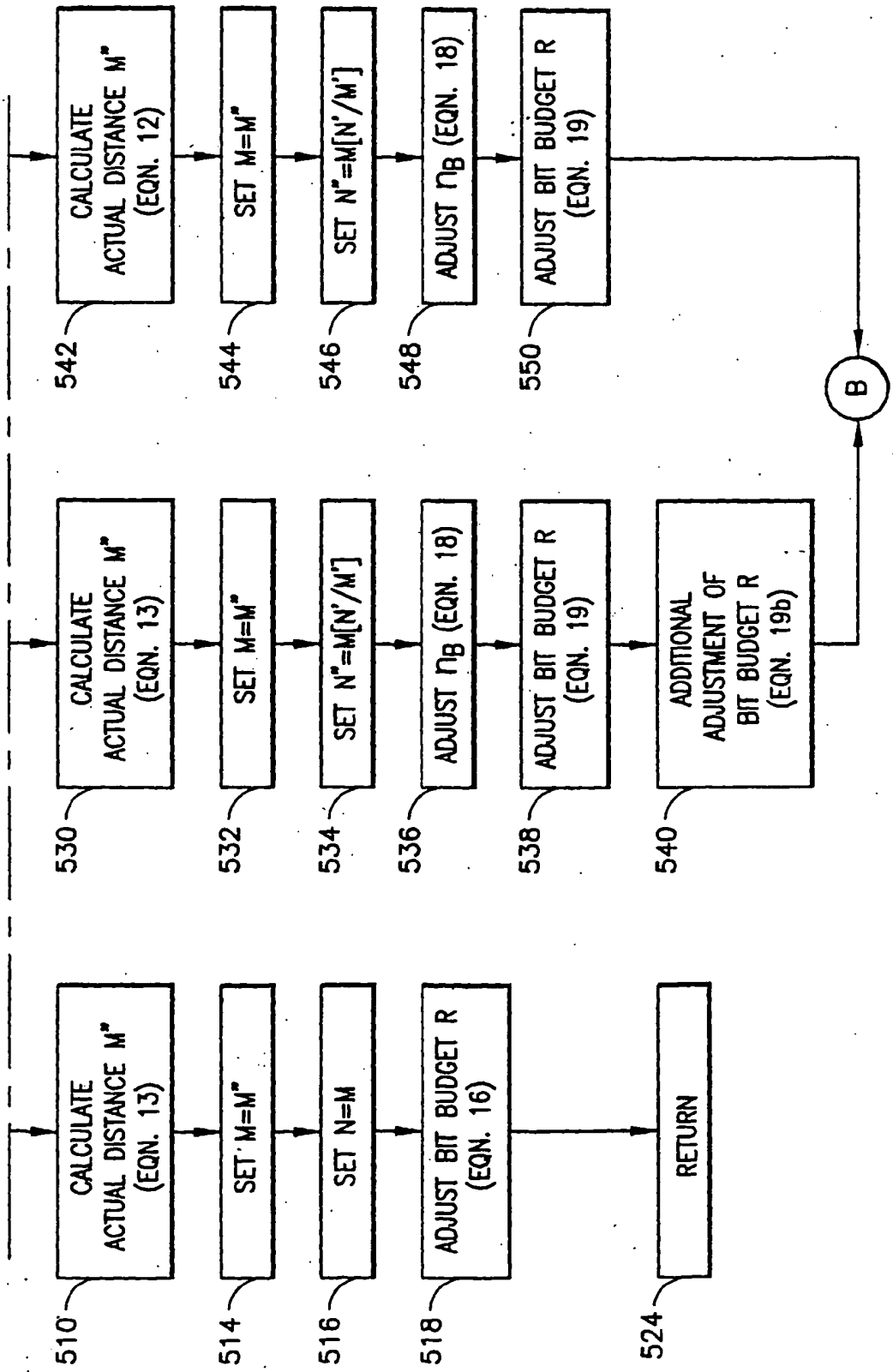

CALCULATE ACTUAL DISTANCE M" (EQN. 13)
SET M=M"
SET N=M
ADJUST BIT BUDGET R (EQN. 16)
RETURN
CALCULATE ACTUAL DISTANCE M" (EQN. 13)
SET M=M"
SET N"=M[N'/M']
ADJUST n_B (EQN. 18)
ADJUST BIT BUDGET R (EQN. 19)
ADDITIONAL ADJUSTMENT OF BIT BUDGET R (EQN. 19b)
CALCULATE ACTUAL DISTANCE M" (EQN. 12)
SET M=M"
SET N"=M[N'/M']
ADJUST n_B (EQN. 18)
ADJUST BIT BUDGET R (EQN. 19)
B
510
514
516
518
524
530
532
534
536
538
540
542
544
546
548
550

FIG.4b-2

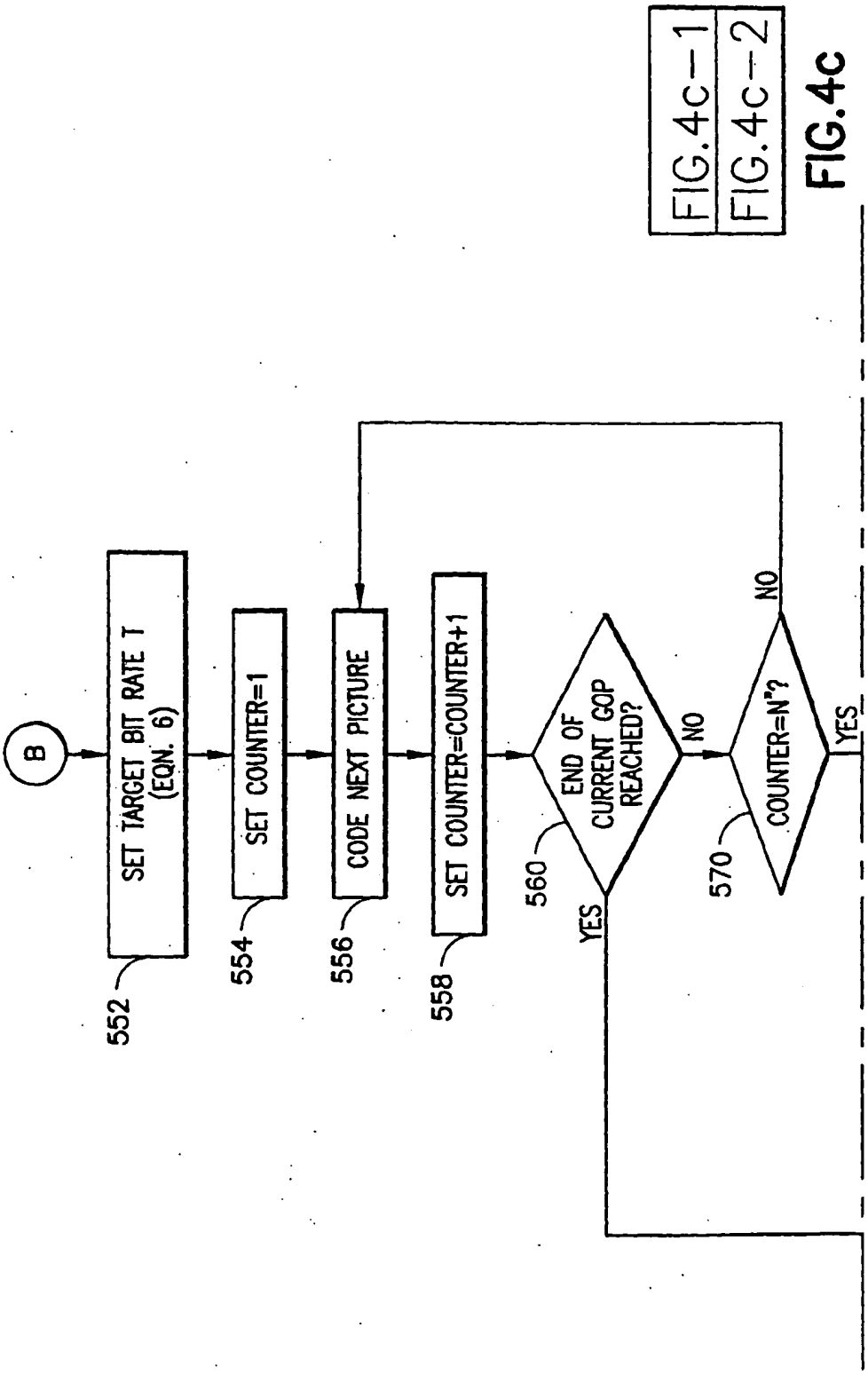


FIG.4c-1

FIG.4c

FIG.4c-1
FIG.4c-2

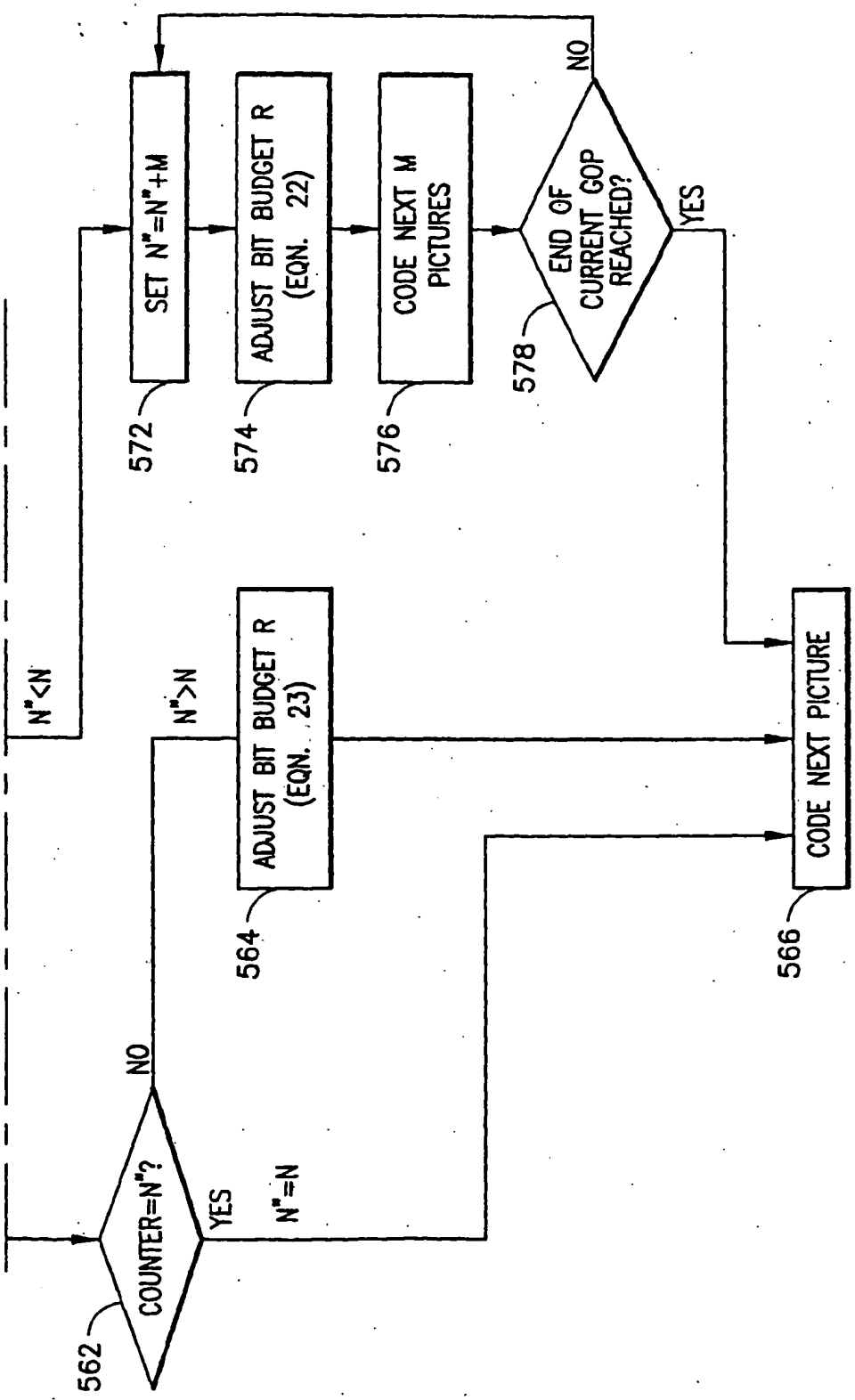

COUNTER=N"?
562
NO
YES
N"=N
N"<N
N"=N"+M
572
ADJUST BIT BUDGET R (EQN. 22)
574
CODE NEXT M PICTURES
576
END OF CURRENT GOP REACHED?
578
NO
YES
N">N
ADJUST BIT BUDGET R (EQN. 23)
564
CODE NEXT PICTURE
566

FIG.4c-2

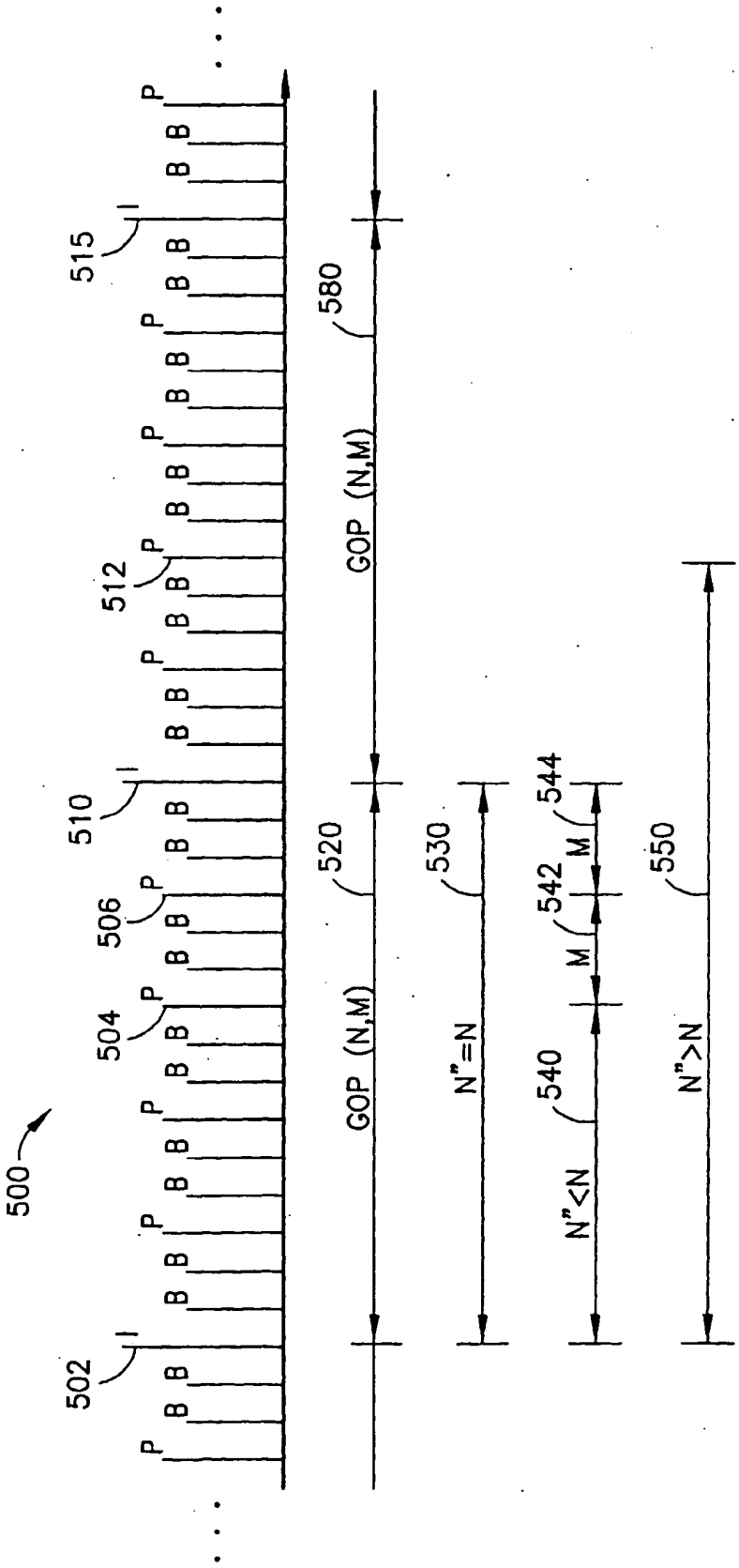

500
502
504
506
510
512
515
520
530
540
542
544
550
580
GOP (N,M)
GOP (N,M)
N"=N
N"<N
N">N
M
M

FIG.5

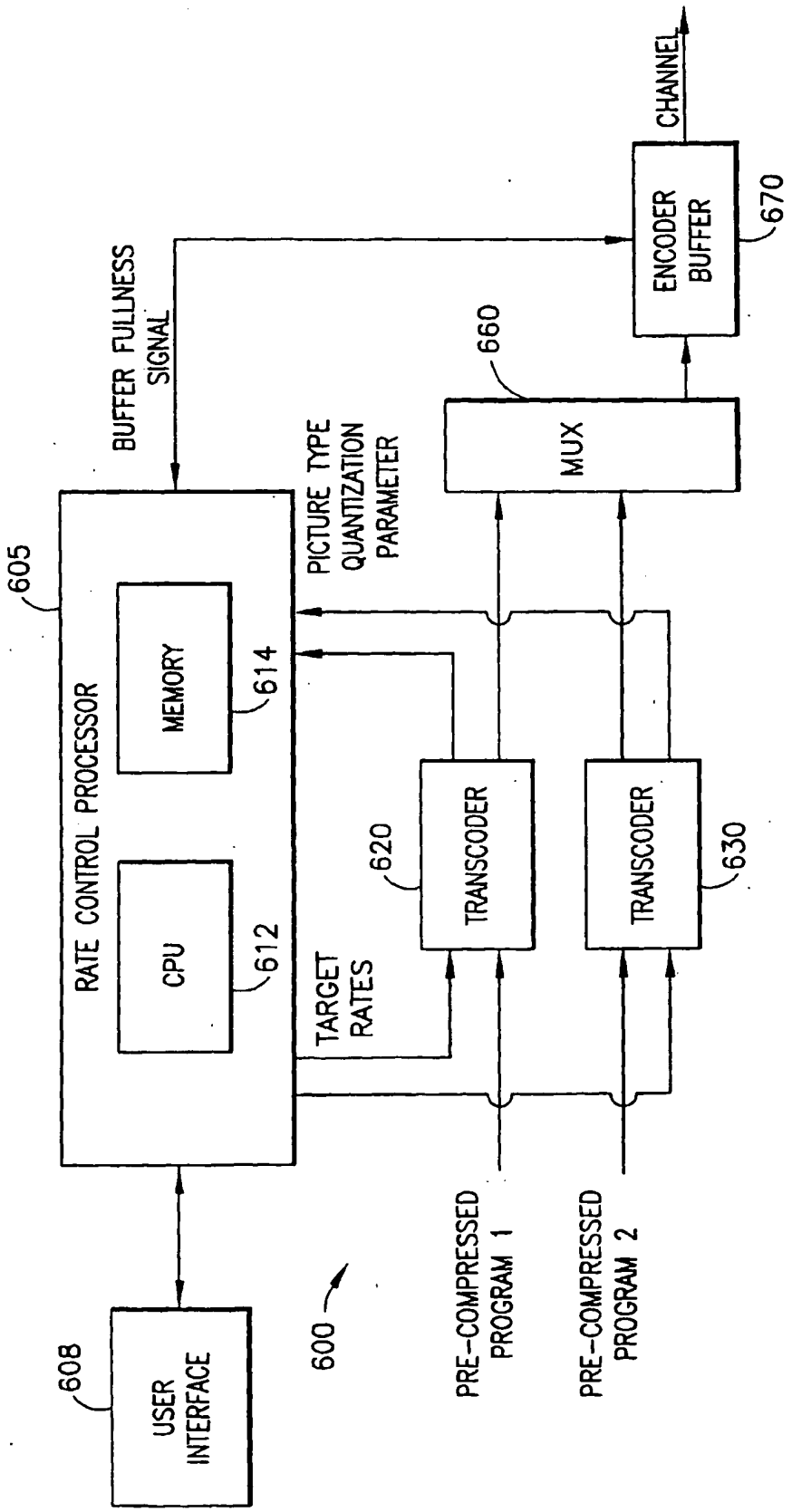

CHANNEL
ENCODER BUFFER
670
660
MUX
BUFFER FULLNESS SIGNAL
PICTURE TYPE QUANTIZATION PARAMETER
605
MEMORY
614
RATE CONTROL PROCESSOR
CPU
612
620
TRANSCODER
630
TRANSCODER
TARGET RATES
600
608
USER INTERFACE
PRE-COMPRESSED PROGRAM 1
PRE-COMPRESSED PROGRAM 2

FIG.6

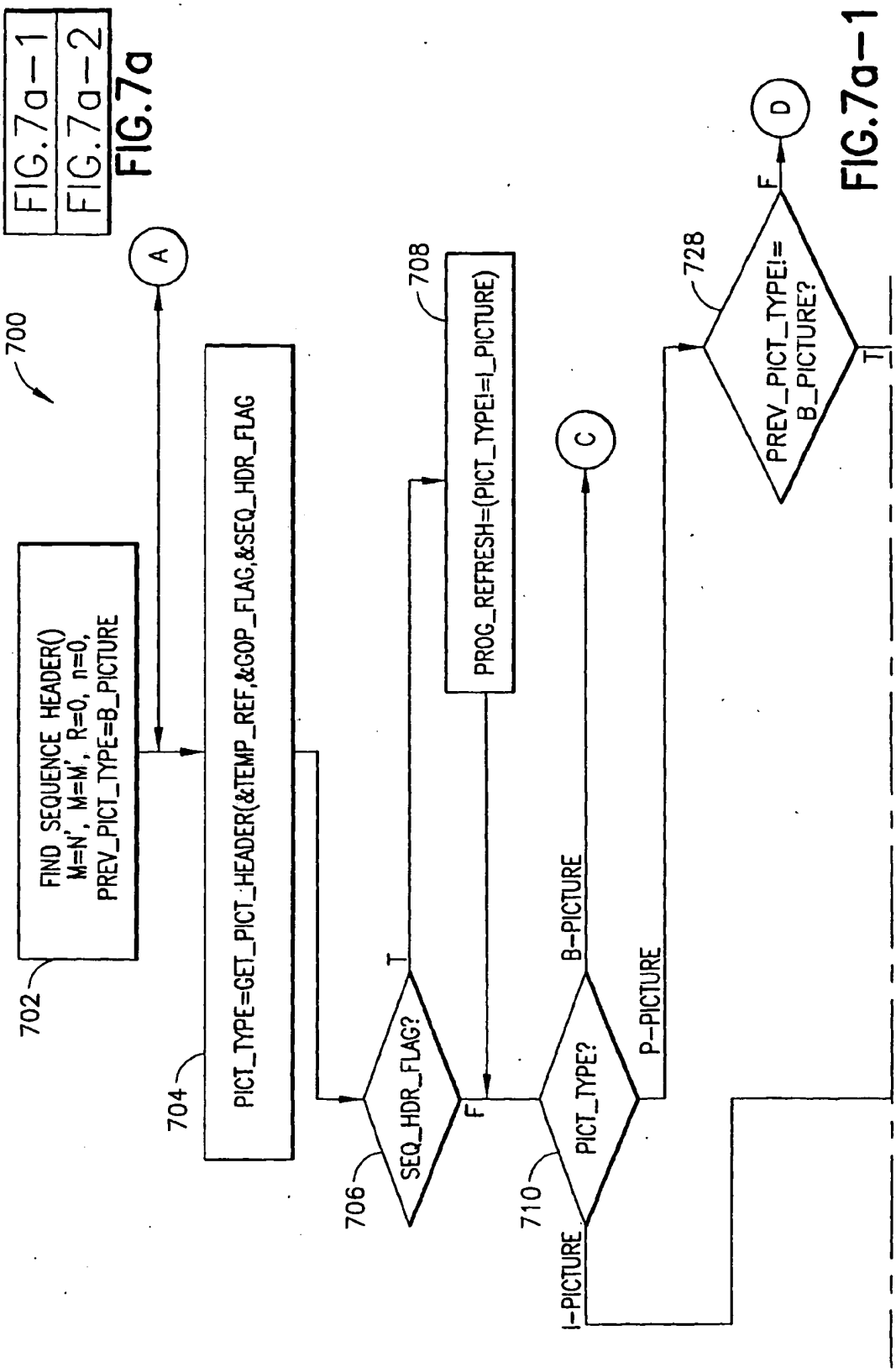


FIG.7a-1
FIG.7a-2
FIG.7a
700
FIND SEQUENCE HEADER()
M=N', M=M', R=0, n=0,
PREV_PICT_TYPE=B_PICTURE
702
A
PICT_TYPE=GET_PICT_HEADER(&TEMP_REF,&GOP_FLAG,&SEQ_HDR_FLAG
704
SEQ_HDR_FLAG?
706
T
F
PROG_REFRESH=(PICT_TYPE!=I_PICTURE)
708
PICT_TYPE?
710
B-PICTURE
P-PICTURE
I-PICTURE
C
PREV_PICT_TYPE!=
B_PICTURE?
728
F
T
D
FIG.7a-1

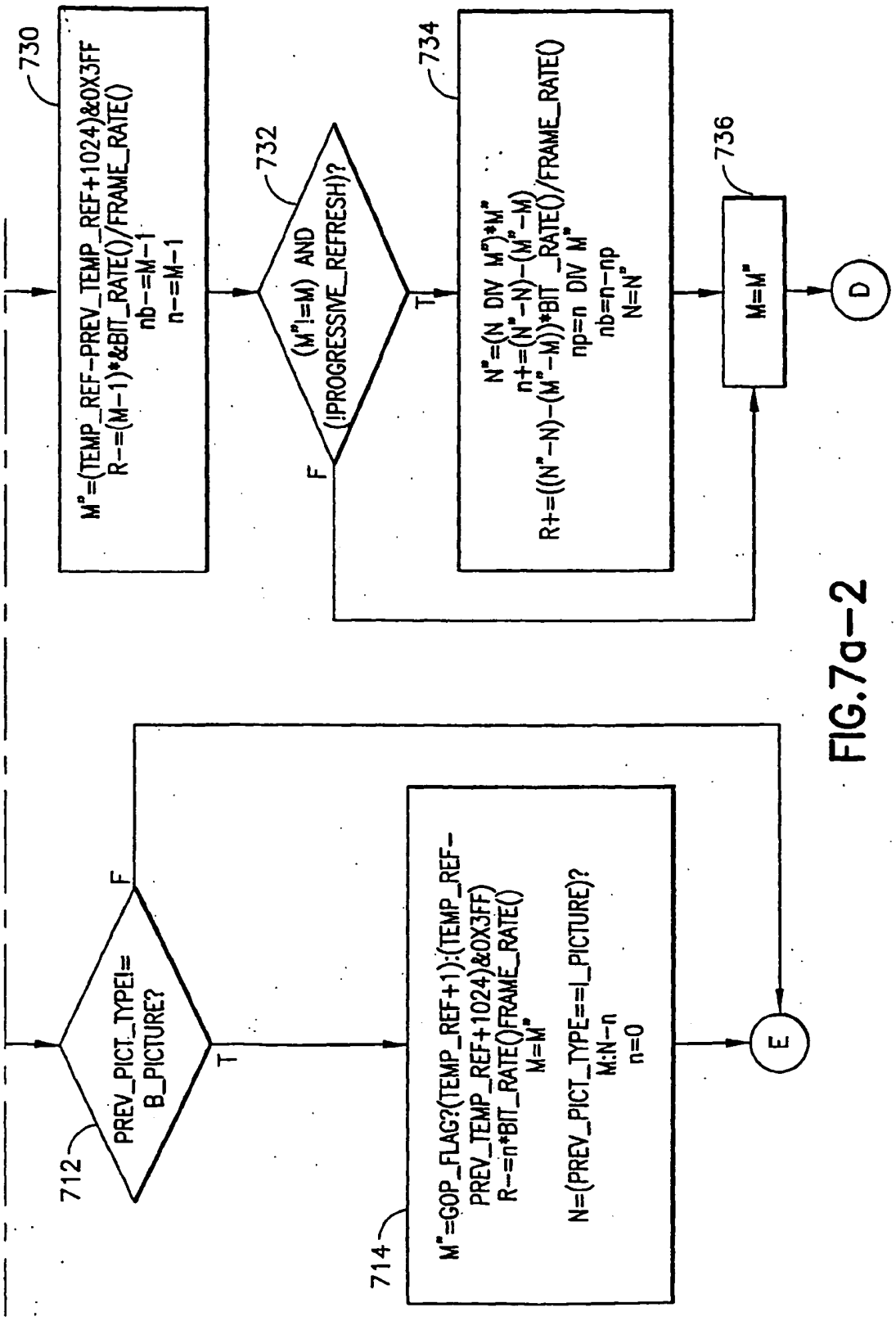

712
PREV_PICT_TYPE!=
B_PICTURE?
T
F
714
M"=GOP_FLAG?(TEMP_REF+1):(TEMP_REF-
PREV_TEMP_REF+1024)&0X3FF)
R-=n*BIT_RATE()/FRAME_RATE()
M=M"
N=(PREV_PICT_TYPE==I_PICTURE)?
M:N-n
n=0
730
M"=(TEMP_REF-PREV_TEMP_REF+1024)&0X3FF
R-=(M-1)*&BIT_RATE()/FRAME_RATE()
nb-=M-1
n-=M-1
732
(M"!=M) AND
(!PROGRESSIVE_REFRESH)?
T
F
734
N"=(N DIV M")*M"
n+=(N"-N)-(M"-M)
R+=((N"-N)-(M"-M))*BIT_RATE()/FRAME_RATE()
np=n DIV M"
nb=n-np
N=N"
736
M=M"
D
E

FIG.7a-2

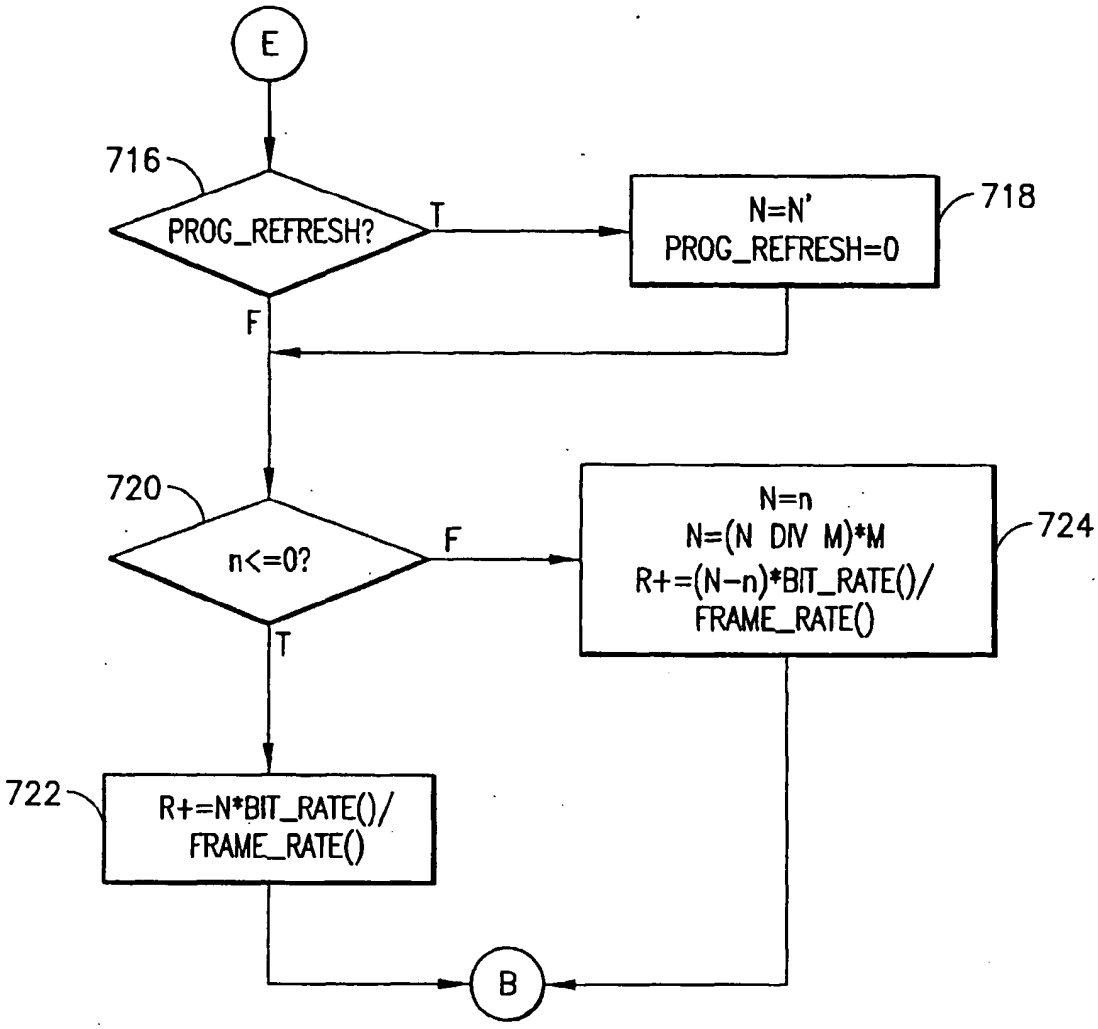

E
716
PROG_REFRESH?
T
N=N'
PROG_REFRESH=0
718
F
720
n<=0?
F
N=n
N=(N DIV M)*M
R+=(N-n)*BIT_RATE()/
FRAME_RATE()
724
T
722
R+=N*BIT_RATE()/
FRAME_RATE()
B

FIG.7b

FIG.7c

FIG.7c-1
FIG.7c-2

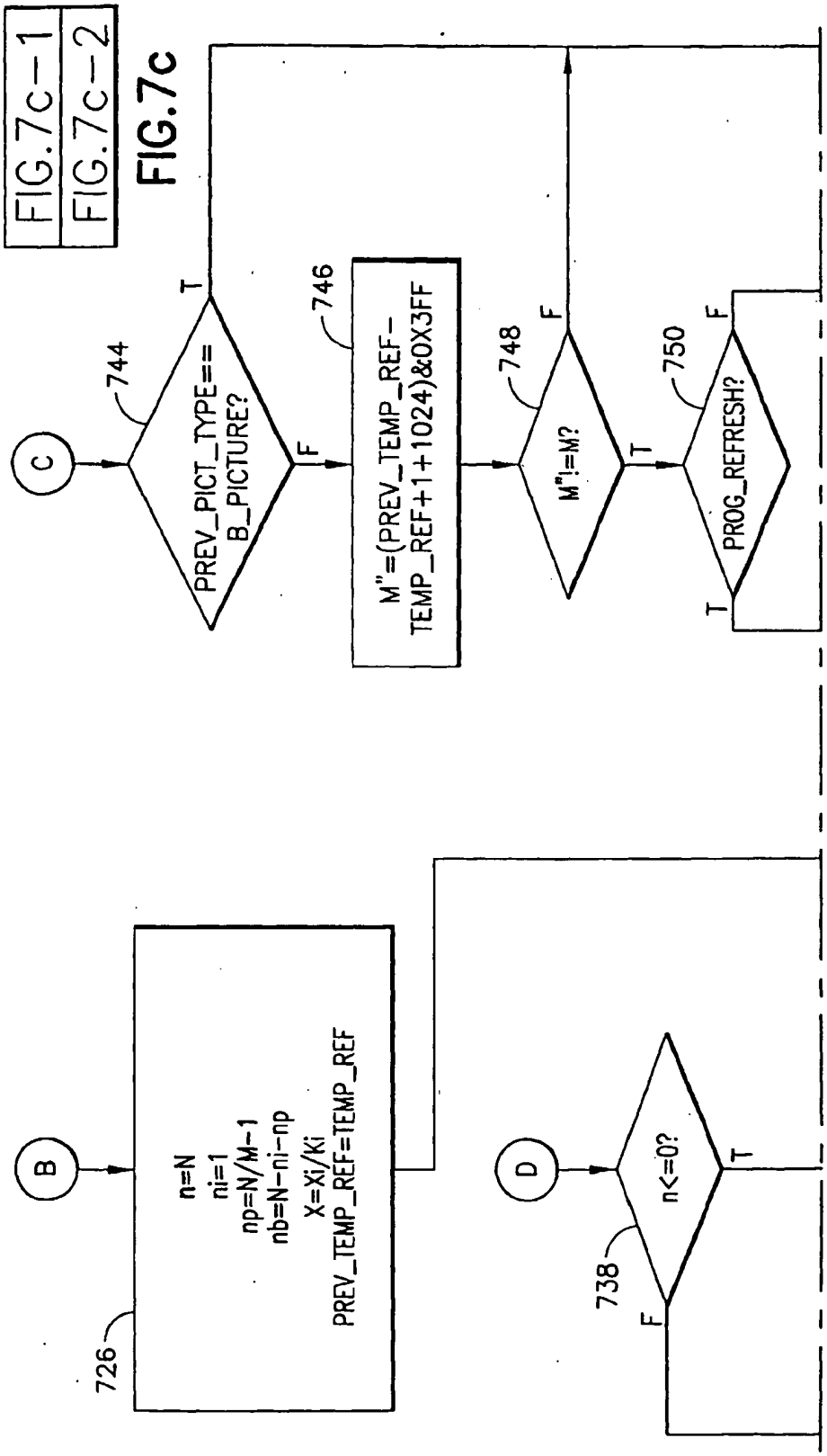


FIG.7c-1

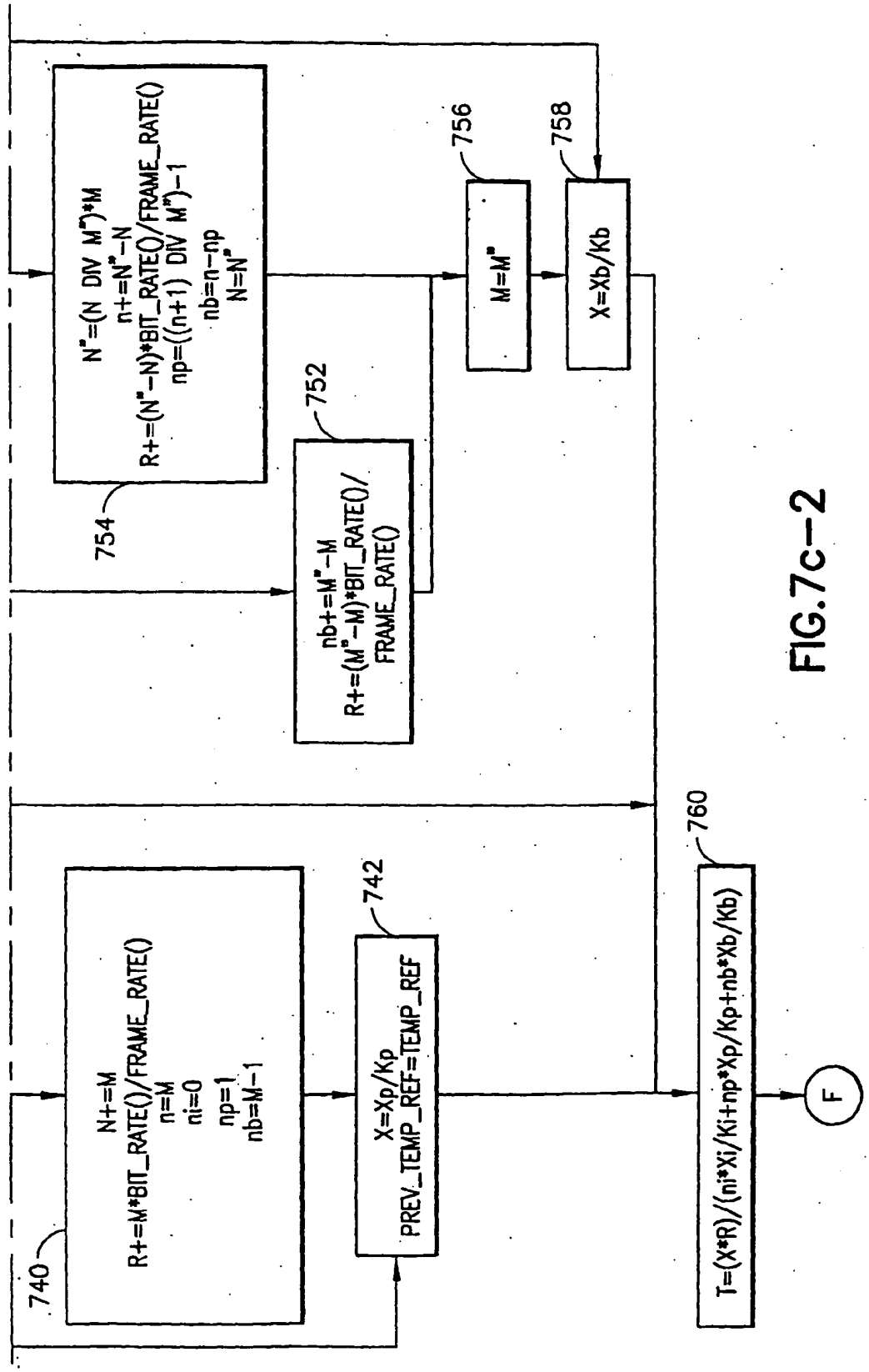

740
N+=M
R+=M*BIT_RATE()/FRAME_RATE()
n=M
ni=0
np=1
nb=M-1
742
X=Xp/Kp
PREV_TEMP_REF=TEMP_REF
754
N"=(N DIV M")*M
n+=N"-N
R+=(N"-N)*BIT_RATE()/FRAME_RATE()
np=((n+1) DIV M")-1
nb=n-np
N=N"
752
nb+=M"-M
R+=(M"-M)*BIT_RATE()/
FRAME_RATE()
756
M=M"
758
X=Xb/Kb
760
T=(X*R)/(ni*Xi/Ki+np*Xp/Kp+nb*Xb/Kb)
F

FIG.7c-2

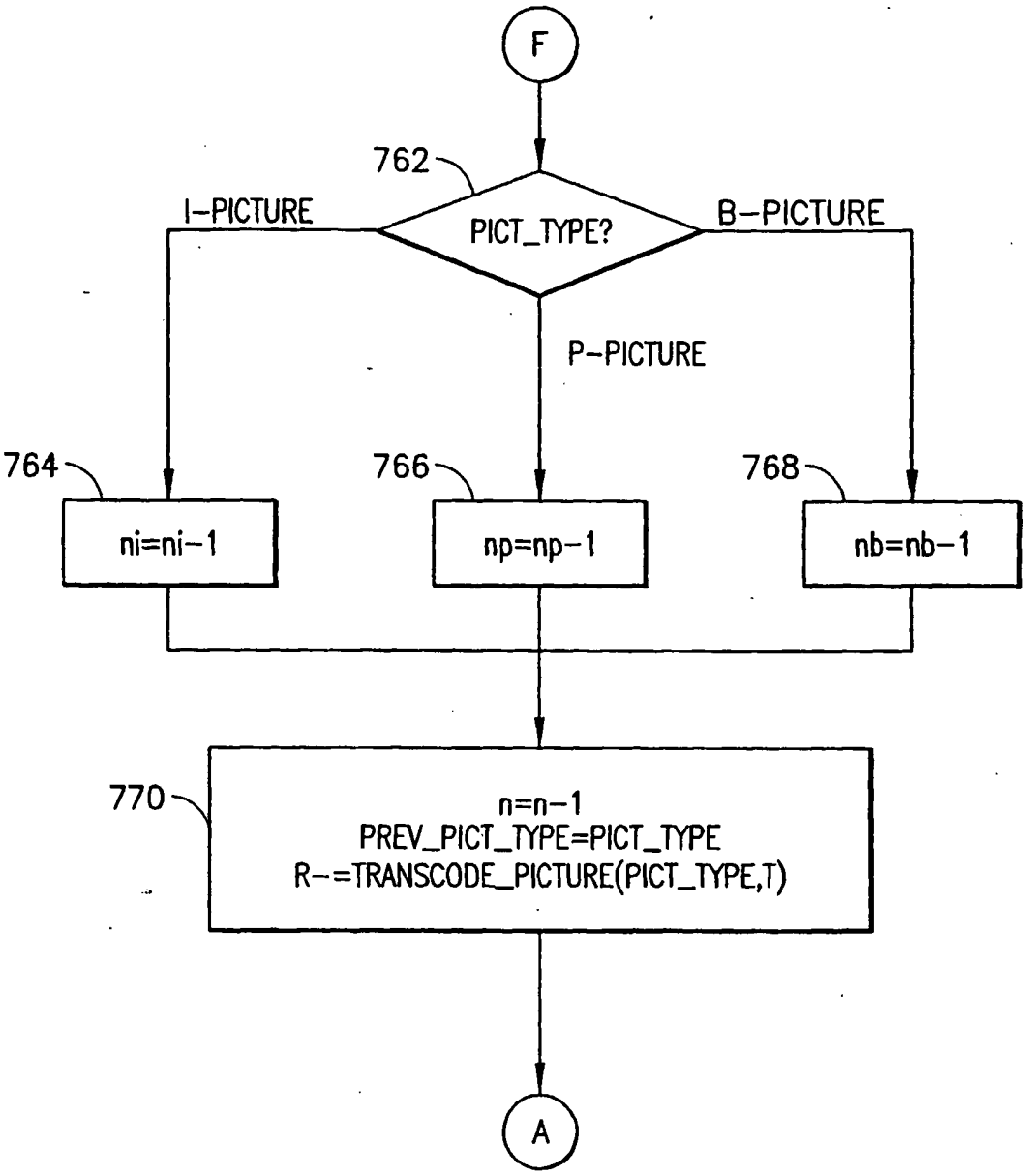

F
762
PICT_TYPE?
I-PICTURE
P-PICTURE
B-PICTURE
764
766
768
ni=ni-1
np=np-1
nb=nb-1
770
n=n-1
PREV_PICT_TYPE=PICT_TYPE
R-=TRANSCODE_PICTURE(PICT_TYPE,T)
A

FIG.7d

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG

Document: **JVT-E146d37**
File: JVT-E146d37ncm.doc
Generated: 2002-11-17

5th Meeting: Geneva, Switzerland, 9-17 October, 2002

Title: Editor's Proposed Draft Text Modifications for Joint Video Specificati on (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), Geneva modifications draft 37

Status: Draft Output Document of JVT

Contact: Thomas Wiegand
Heinrich Hertz Institute (HHI), Einsteinufer 37, D-10587 Berlin, Germany
Tel: +49 - 30 - 31002 617, Fax: +49 - 030 - 392 72 00, wiegand@hhi.de

Purpose: Report

[Ed. Notes:

Modifications in JVT-E146d0 relative to JVT-E022d7:

- Mike Nilsson's edits on *decoded picture* buffer
- Mike Nilsson's edits on SP frames
- Mike Nilsson's edits on intra_pred_mode
- Mike Nilsson's edits on CAVLC
- Input on Mb-adaptive frame-field

Modifications in JVT-E146d1 relative to JVT-E146d0:

- minor edits and field updates

Modifications in JVT-E146d2 relative to JVT-E146d1:

- incorporation of text regarding the adopti on of JVT-E088

Modifications in JVT-E146d3 relative to JVT-E146d2:

- From JVT-E152r1

Changes related to HRD (JVT-E075, all except item 1):

- Changed name of "decoded pictures buffer (DPB)" to "decoded picture buffer (DPB)" to eliminate conflict of DPB with "slice data partition B." Added DPB acronym.
- Changed name of "pre-decoder buffer" to "coded picture buffer (CPB)," which was already in the list of definitions.

  - pdb_cnt -> cpb_cnt

- Changed name of "pre_dec_removal_delay" to "cpb_removal_delay."
- Changed name of "initial_pre_dec_removal_delay" to "initial_cpb_removal_delay."
- Removed references to "VBV."
- Changed name of "HRD Picture SEI Message" to "Picture Timing SEI Message"
- Two Ed.Notes in D.3.9 about random access definition. *Gary says the issue is real and we should fix it by eliminating the preroll thingie.*
- Changed semantics of "cpb_removal_delay" so that the delay is now with respect to the previous random access point rather than the previous picture. The "entropy coding" of the syntax element is al so changed to be a fixed length code of specifiable length. *Random access point is not clearly defined.*
- Added language specifying that the "initial_cpb_removal_delay" must not exceed the time -equivalent of the CPB size.
- Changed name of "coded_pic_buffer_ size_value" to "cpb_size_value" and "coded_pic_buffer_size_scale" to "cpb_size_scale."

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

- 3.3.6 modify description of IDR picture's action on decoded picture buffer
- Change 8.3.6.7.6 "reset" to "reset_reference" add "reset_nonreference" and "reset_all"

EDITORIAL

- Change "reference picture" to "decode d picture" in several places
- Sec 7.3.3.3 change ref_ to dec_

Modification to support frame store implementations and unify to single decoded picture buffer JVT -E034

TECHNICAL (NOT INCORPORATED !!!!!!)

- A new field max_sliding_window_size_minus_one added to sequence parameter set
- Define operation of the sliding window supporting frame stores and consistent with a single decoded picture buffer.
- Operation defined to support frame storage. Dangling fields occupy an entire frame store.

EDITORIAL

- Modify definition of decoded picture buffer
- Remove definition of reference picture buffer
- Replace reference picture buffer with decoded picture buffer.
- Replace mention of long-term and short-term buffers describing elements of decoded picture buffer
- Add "reference" before "frame" as needed
- Modify description of constructing picture numbers for field indexing or frame indexing.
- Replace " max_long_term_idx_plus1" with "max_long_term_pic_idx_plus1"
- max_long_term_idx_plus1 MINUS 1 is the largest long -term frame index

INCLUDED RESPONSE to comments raised during review.

- Replaced "decoded pictures buffer" with "decoded picture buffer"
- Modified discussion in Annex A to reflect use of single decoded picture buffer simultaneously included text for HRD DPB verifier since this is tied to the discussion modified in Annex A.
- Note the HRD DPB was added to this document since it references the modifications to Annex A

- Delete "as un-used" in definition of IDR.
- Correct check for while loop in Decoded picture buffer management syntax
- Change non-reference_pictures_reset_flag and reordering_pictures_reset_flag to non_reference_pictures_reset_flag
- Change text that were not syntax nor variable names to English
    - Reset_all → reset all referece and non-reference pictures
    - Reset_reference → reset all reference pictures
    - Reset_non-reference → reset all non-reference pictures
- Clean up some text and spelling

Modifications in JVT-E146d5 relative to JVT-E146d4:

Editorial from TK's e-mail

- removed redundent definition in IDR picture.
- replaced definition of 1 picture with 1 slices.
- Indentation for dec_pic_buffer_management( ) in the syntax table
- Changed non_reference_pictures_reset_flag to non_ref_pic_reset_flag
- Changed remaining ref_pic_..... syntax element to dec_pic_.....
- Add missing references to Unused pictu re marking

Modifications in JVT-E146d6 relative to JVT-E146d5:

Technical from JVT-E156

- reduction of sample dependencies for 4x4 intra prediction (JVT -E111)

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

- change of direct temporal mode to 8x8 MC blocks (JVT -E037)
- removal of special position (JVT-E161)

Moved switch from slice header to switch at Sequence Parameter Set of special 8x8 MC block mode

Modifications in JVT-E146d7 relative to JVT-E146d6:

- Moved S pictures into clause 8
- Removal of ABT
- Removal of Intra_8x8
- Removal of 1/8-pel MC

Modifications in JVT-E146d8 relative to JVT-E146d7:

Outstanding items from JVT-E135

- Suggested CAVLC semantics
- Editorial correction and clarifications
- Addition of chroma_qp_index_offset

Modifications in JVT-E146d10 relative to JVT-E146d8

- Moved the table 9.3 to subclause 7.4. 4
- Removed Mapped Golomb code from clause 9
- Change me(v) to ue(v) in Syntax table
- Change xe(v) to te(v) in Syntax table
- Delete definition of me(v)
- Changed coded_block_patternY to CodedBlockPatternLuma
- Changed nc to CodedBlockPatternChroma
- Change diagram for macroblock to remove CBPY.
- Change code_num to CodeNumber
- Change field and frame reference for ref_idx_10 text into tables.

Modifications in JVT-E146d11 relative to JVT-E146d10

- incorporation of text regarding of JVT -E086
- PCM mode for CABAC
- incorporation of text regarding to JVT-E059
- removal of CABAC text related to ABT
- modifications for removal of intra_8x8 sub MB type
- modifications of init tables
- incorporation of text regarding of JVT -E154

Modifications in JVT-E146d12 relative to JVT-E146d11

- Replace num_of_ref_frames -> num_ref_frames
- nal_storage_idc should be restricted to the same value for all slices in a picture – Strike that, it should be restricted such that it is either zero for all slices in a picture or non-zero for all slices in a picture, but not necessarily the same non-zero value for all NAL units of a picture. – This is stated already in the semantics of nal_reference_idc.
- Editorial correction of default list 0 and list 1 reference indices for B slices. List 0 orders past pictures from most recent to oldest, the future pictures from current time towards the future, then long -term pictures. List 1 orders the future pictures from current time towards the future past, then past pictures from most recent to oldest, then long-term pictures.
- Added text to define more precisely the default orders for field pictures. Default order is as frame order, but is ordered same parity followed by opposite parity.

TECHNICAL changes:

- Added two fields to VUI HRD parameters specifying the length of the cpb_removal_delay syntax elements.
- Removed references to "prev_buf_period_duration."

Changes related to temporal reference (JVT-E122 and JVT-E078):

- Defined "dpb_output_delay" in the Picture Timing SEI Message to replace "temporal_ref_value." Added its sementics. Added description of initial and final DPB output times.
- Added one fields to VUI HRD parameters specifying the length of the dpb_output_delay syntax element.
- Added "output _type" syntax and semantics to the Picture Timing SEI Message.
- Added "field_delay_value" to the Picture Timing SEI Message.
- Added two more fields to VUI HRD parameters specifying the length of the dpb_output_delay and field_delay_value syntax elements.

Changes related to film mode support (JVT-E078):

- Changed "pic_structure" to "field_pic_flag" and "bottom_field_flag" as appropriate.
- Modified several syntax tables to use the new elements.

Changes related to clock timestamp support (JVT-E122):

- Added "ct_type" and "nuit_field_based_flag" to the clock timestamp SEI message (syntax and semantics).
- Modified Equation D-1 to use nuit_field_based_flag.
- Moved syntactical constraints out of Annex C.

Modifications in JVT-E146d0ncm_hrd_edits_r5 relative to JVT-E146d0ncm_hrd_edits:

- Removed some references to "packet stream" in HRD description.
- Removed *more* references to "prev_buf_period_duration."
- change "hrd_picture" to "pic_timing" in two pl aces
- combined the Clock Timestamp SEI message and the HRD Picture message into a single Picture Timing SEI message.

Changes related to modified CAT proposal (JVT-E133):

- Added syntax element "initial_cpb_removal_delay_offset."
- Added description of "initial_cpb_removal_delay_offset" effect on HRD operation in Annex C.1 and informative text in C.2.

Changes related to clock timestamp support (JVT-E122):

- Renamed "output_type" to "pic_structure"
- Placed clock timestamp syntax inside pic_structure if statement, and allowed multiple copies of the clock timestamp syntax elements.
- Removed reserved bits.
- Removed two no longer existent SEI payload types.
- Added semantics for equal EquivalentTimestamps.

Changes related to clock timestamp support (JVT-E032):

- Added description of DPB operation

Modifications in JVT-E152r1 relative to JVT-E152:

- Changed description of CPB removal time in C.1.1.2 to accurately reflect the adoptions in this meeting.

Various changes based on input from reviewers of JVT-E152.

Modifications in JVT-E146d4 relative to JVT-E146d3:

From JVT-E149r1

Modifications for JVT-E121 (Incorpoation of TK's revisions)

TECHNICAL

- 7.4.3.2 change mmco command "reset" to "reset_reference", add commands "reset_nonreference" and "reset_all" modify and add descriptions
- Table 7-9 modify entry 5 add entries 6 and 7

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

- 8.3.6.6 Sliding window operation
- A.6.1 must conform to DPB verifier with capacity determined by the level
- A.6.2 replaced reference memory with decoded picture buffer memory
- A.6.3 Defined capacity of DPB in frames used in HRD
- C.1.2.(1-4) Reorder and modify text of operations: Decode, Mark, Remove, Add. Conformance is only tested in addition step this ordering simplifies the description.
- C.1.2.5 Conformance references DPBcapacity calculated in A.6.3

Modifications in JVT-E146d13 relative to JVT-E146d12

- Edits on MB-AFF and Picture Order Codes in Chapters 7, 8, and 10.
- Also a statement that successive IDR pictures are forbidden.
- SEI DPB output times shall have the same ordering as Picture Order Counts.

Modifications in JVT-E146d14 relative to JVT-E146d13

- limitation of cbp_cnt to be no greater than 32 (shouldn't we be using cpb_cnt_minus1 inst ead of cpb_cnt with a value of 0 being prohibited?) from JVT-E123
- limitation of entropy_coding_mode to not exceed 1 from JVT-E123 [This constraint then disappeared somewhere between draft 16 and draft 22 and was put it back in draft 25.]
- increased the maximum number of sequence parameter sets to 31 from JVT-E123
- increased the maximum number of picture parameter sets to 128 from JVT-E123
- removed non_stored_content_flag and non_stored_pic_flag redundant syntax element from JVT-E123
- moved deblocking_filter_parameters_present_flag and constrained_intra_pred_flag from the sequence parameter set to the picture parameter set from JVT-E123
- removed redundant entries in sample aspect ratio table and made sar_width and sar_height u(16)'s. from JVT-E123
- removed motion_resolution from JVT-E123
- made frame_cropping_flag and timing_info_flag u(1) instead of ue(v) from JVT.-E123 [The frame_cropping_flag change part of this was somehow missed and was put back in draft 25.]
- changed some high-level syntax elements from e(v) to ue(v) from JVT-E123

Modifications in JVT-E146d15 relative to JVT-E146d14

- from JVT-E146d10AG.doc
- AG's paper edits

Modifications in JVT-E146d16 relative to JVT-E146d15

- from JVT-E159 changes for alternate scan
- from JVT-E162 changes regarding deblocking filter:
  - Figure 8-14 was updated to deal with the MB-AFF case. Text was added to explain that
  - Text added to explain how to compute QPav for the deblocking filter in the case where any of the blocks if composed of fields with different QP

Formatted: Indent: Left: 0.8", Bulleted + Level: 1 + Aligned at: 0.55" + Tab after: 0.8" + Indent at: 0.8", Tabs: Not at 0.8"

- from JVT-E154r4
  - "long" and "short" SCPs removed – there is now only one SCP length (3 bytes), but sometimes an extra byte of value 0 is stuffed before it. (per meeting)
  - JVT-E054 incorporated. (change to 3-byte short SCPs from 2-byte short SCPs, send (what was) Long SCP only on Parameter Sets and first slice of each picture)
  - Specified that reserved NAL unit types shall be ignored by decoders.
  - Changes to maximum bit rates and buffer capacities (esp. levels 3 to 5), specify VCL = 1, VCL+NAL = 1.2. [Do we have a clear definition of wh at counts in VCL and what counts in NAL?]
  - Removal of field pictures (all interlace-oriented decoding tools) in baseline profile
  - Updated informative table in Annex A to reflect new Levels table values.

Formatted: Indent: Left: 0.8", Bulleted + Level: 1 + Aligned at: 0.55" + Tab after: 0.8" + Indent at: 0.8", Tabs: Not at 0.8"

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

- Reworded the HRD semantics for xxx_hrd_parameters_present_flag to reflect the new bitstrea m conformance requirements.
- Moved the semantics for low_delay_hrd into the correct location in the sequence VUI subclause.
- Minor edits to first three paragraphs of C.1.
- Two new paragraphs in C.1 explaining the operation of the HRD and the various locations of HRD-related syntax and semantics.
- Minor edits to C.1.1
- Added definition of "random access point" to the list of definitions.
- Corrected range of index values for chroma DC coefficient scaling equations (USNB subject 25)

Modifications in JVT-E146d26 rela tive to JVT-E146d25

- Swapped RBSP trailing bits and RBSP slice trailing bits syntax and semantics sections with each other (this way everything that uses RBSP trailing bits precedes the definition of that structure rather than some being before and some aft er)
- Clarification of byte-stream and cabac_stuffing_word issues.
- Range of values specifications for weighted prediction syntax elements.
- Addition of editor's notes regarding need for clarification of issues surrounding MB -AFF.
- Clarification of calculation of motion vector components and reference picture indices.
- Clarification of B-picture issues.
- Clarification of weighted prediction issues.
- Added overscan_info_present_flag and overscan_appropriate_flag to sequence VUI per JVT -E055.
- Removed HRD picture SEI message syntax (was apparently left in by accident, although the semantics had been removed).
- Removed clock timestamp SEI message syntax (was apparently left in by accident, although its content had been subsumed into the picture timing SEI message).
- Added maximum output rate constraint to Section C.1.2.4. Updated C.1.1.3 maximum removal rate definition. Now both limits are in terms of the maximum macroblocks per second (A.6.2) and the number of macroblocks in the picture. Definition works for fields. F rom JVT-E123.
- Aligned the numbering of the subclauses 8.5.3.1 -4 for the four chroma intra prediction modes with the order of the modes in the VLC table 7-15. At request of USNB Subject 15.
- Added new SEI messages from JVT-E038 (full-frame freeze & release, full-frame snapshot, progressive refinement segment start & end) and cleaned up SEI payload type syntax to remove old deleted messages.
- Swapped the order of buffering period SEI message and picture timing SEI message in syntax and semantics (since buffering period is how you start the HRD process and picture timing is how you then continue it).

Modifications in JVT-E146d27 relative to JVT-E146d26

- Correction of Figure 8-5 – Differential motion vector decoding with scaled motion vector.
- Removed motion vector clipping to +/- 19 samples off edge of picture per USNB subject 14.
- Clarification of nal_storage_idc, including renaming of nal_storage_idc to nal_reference_idc.
- Clarification and bounding of ranges of sub -sequence SEI messages.
- Reserved a range of values of pan_scan_rect_id, progressive_refinement_id, and snapshot_id for future use by ITU-T | ISO/IEC.

Modifications in JVT-E146d28 relative to JVT-E146d27

- Added editorial note re problem in Figure 9-6 for CABAC.
- Corrected variable name "symbol" to "S" in sub clause 9.2.1.3.
- Redefinition and relaxation of minimum compression ratio requirement per USNB subject 32.
- Clarification of clock_timestamp per USNB subject 40.
- Clarification of broken_link_flag per USNB subject 42.
- Clarification and bounding of ranges for pan-scan rectangle SEI messages.
- Correction of CPB buffer size for Level 5.

- from JVT-E146d14ncm_gjs
  - added max_dec_frame_reordering and constrained its range of values
  - constrained the range of values for the frame cropping rectangle.
  - constrained the range of the maximum motion vector length parameters
  - clarified and constrained the range of the syntax elements that limit the number of b its per picture and the number of bits per MB
- from JVT-E146d7_hrd_hrd
  - E.3.2 added text indicating limits on bit_rate[k] and cpb_size[k].
  - D.3.1 Added details and clarifications.
  - For the second time, the obsolete syntax element removal_time_tolerance is rem oved from vui_seq_parameters().
  - Edits to preamble part of C.1.1 to reflect agreed HRD conformance status.
  - Figure C-1 updated to replace "bitstream" with byte stream.
  - Fixed cross-references in Annex C; unable to fix all because the Figures C-3 to C-5 do not appear in the list of Figures and can't be properly cross-referenced.
  - Similar problem with the tables in Annex D.
  - NOTE – I believe vcl_hrd_flag should be removed. It is no longer normative to produce a stream for which the vcl does not confrom to the HRD .

Modifications in JVT-E146d17 relative to JVT-E146d16

- from JVT-E165r3
- integrated the B picture clause into subclause 8.4
- from JVT-E146d16ncm_e60

Modifications in JVT-E146d18 relative to JVT-E146d17

- from JVT-E146d16_jill
- from JVT-E146d16AMT
- Changed POD to DiffPicOrderCount: Difference of Picture Order Counts

Modifications in JVT-E146d19 relative to JVT-E146d18

- from JVT-E157
  - from JVT-E124 changes for
  - Adding frame_coding_only_flag into sequence parameter set, to skip signalling field_pic_flag and bottom_field_flag in slice header.
  - IDR picture is always stored as a reference picture
  - Restricitions on slice header parameter values
  - Picture order count mode 2 (done already in JVT-E146d16)
  - from JVT-E125 changes for
  - Picture boundary detection based on field parity (done already in JVT-E146d16)
  - from JVT-E126 changes for
  - Clarification and bug fixes of the sub-sequence SEI messages
  - from JVT-E128 changes for
  - Moving mb_slice_group_map_type 2 to the last order (6), then types 3~6 become 2~5.
  - from JVT-E130 changes for

Formatted: Indent: Left: 0.8", Bulleted + Level: 1 + Aligned at: 0.55" + Tab after: 0.8" + Indent at: 0.8", Tabs: Not at 0.8"

Formatted: Indent: Left: 0.8", Bulleted + Level: 1 + Aligned at: 0.55" + Tab after: 0.8" + Indent at: 0.8", Tabs: Not at 0.8"

Formatted: Indent: Left: 0.83", Bulleted + Level: 1 + Aligned at: 0.55" + Tab after: 0.8" + Indent at: 0.8", Tabs: Not at 0.8"

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

- Replacement of Figure C-5 (some kind of problem had made it not visible)

Modifications in JVT-E146d29 relative to JVT-E146d28

- Clauses 8 and 9 describe processes or provide a specif ication for the processes -> changed all clause and subclause names to contain the phrase "process" or "specifaication ... for ... process"
- Added for processes with input/output:

  Input to this process is/are ??? [Ed.Note: specify input to this process]

  Output of this process is/are ??? [Ed.Note: specify output of this process]
- Added for initialisation processes:

  This intitialization process is invoked when ... [Ed.Note: specify]

  Output of this process is/are ??? [Ed.Note: specify output of this process]
- Worked on geometrical concepts
- Clarified exp-Golomb entropy coding.

Modifications in JVT-E146d30 relative to JVT-E146d29

- Added constraint of no more than 16 MVs per two consecutive MBs at levels 3.1 and above.
- Added constraint of 8x8 or larger block size for direct and bi-pred for level 3.1 and above and removed such constraint from level 3 and below.
- Added constraint of no more than 3200 bits per macroblock.
- Added semantics of pcm_alignment_bit and pcm_byte.
- Refined description of exp-Golomb code parsing

Modifications in JVT-E146d31 relative to JVT-E146d30

- Replaced text marked in Finnish with English (UK).
- Clarification of sub-sequence SEI semantics
- Clarification of meaning of decoding process conformance (numerically identical results) in response to USNB Subject 17.
- Added sentence to NAL unit semantics to clarify which NAL units are considered VCL NAL units.
- Clarification of CAVLC parsing process.

Modifications in JVT-E146d32 relative to JVT-E146d31

- Introduced 4 functions in subclause 5.6
- Introduced inverse scanning processes and derivation processes for macroblock and macroblock pair indices in subclause 6.4:
  - Limin, Peter B. - please let me know if these processes work for specifying MB-AFF
  - Limin, Jill, Alexis, Lowell - please let me know whether these processes work for specifying inter prediction
  - Gary - please let me know if these processes work for specifying transform decoding
  - Peter L., Anthony - please let me know if these processes work for specifying deblocking
  - Karl - please let me know if these processes work for specifying CAVLC
  - Detlev - please let me know if these processes work for specifying CABAC
- Rewrote the Intra_4x4 prediction subclause 8.5.1 using the geometric concepts ... first run ... wording could be improved and made less verbose

Modifications in JVT-E146d33 relative to JVT-E146d32

- corrected some MS Word problems
- change data partitioning to slice data partitioning
- removed subclause titles with Introduction or General
- improved subclause 8.3.1 (Intra_4x4)
- changed start of decoding process clause.

Modifications in JVT-E146d34 relative to JVT-E146d33

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

- Adding six definitions of related to redundant slices to clause 3. [Note that numbering of the definitions is not correct and it was not correct in E146d16 either. In addition, some of the definitions are in the wrong places, e.g. "decoded picture buffer mana gement"].
- Change to the semantics of redundant_pic_cnt.
- Bug fix for redundant slices: adding redundant_pic_cnt to headers of slice data partition B and C.
- Change to the semantics of slice_id
- Text changes on 8.2.1 NAL unit decoding order (the subclause is e ntirely modified)
- from JVT-E132 changes for
- Typo correction in 7.3.28 "rsbp" → "rbsp"
- Clarification of mb_slice_group_map_type 3# (old value, the new type number is 2#)
- "MAX_MB_LOCATION " → "MAX_MB_ADDRESS"
- Clarification of scene information SEI message
- from JVT-E141 changes for
- An improvement on the spare picture SEI message: delta_frame_num → target_frame_num
- Bug fix: a "}" was missing

Modifications in JVT-E146d20 relative to JVT-E146d19

- from JVT-E146d16ncm_gjs ... start code clarifications from Gary
- from JVT-E146d15AG ... align names and description for the qp parameters

Modifications in JVT-E146d21 relative to JVT-E146d20

- from JVT-E146d8_aj_deblock
- Changes to macroblock_layer syntax, definition of delta_qp, and deblocking filter interaction with MB AFF from JVT-E162
- Chroma deblocking filter simplifications from JVT -E089
- Clarification and new mode for disable_deblocking_filter_idc from JVT -E123

Formatted: Indent: Left: 0.79", Hanging: 0.3", Bulleted + Level: 1 + Aligned at: 0.55" + Tab after: 0.8" + Indent at: 0.8", Tabs: Not at 0.8"

Modifications in JVT-E146d22 relative to JVT-E146d21

- Some terminology consistency changes (e.g., change "motion comp ensation" to "inter prediction")
- Some changes to POC description (e.g., define same temporal direction based on POC)
- Some fixes relating to MB-AFF and direct prediction and weighted prediction from Boyce, Tourapis, and Wang [Ed. Note: I don't like the use of an ordinary english word with capitalization of the first letter to indicate something special – e.g., "Direct"]

Modifications in JVT-E146d23 relative to JVT-E146d22

- Conversion to automatic numbering for definition and abbreviation numbering
- Conversion to automatic numbering for equation numbering through clause 8 (still need to put in the automatic cross-references).
- Added max_dec_frame_buffering and reworded semantics of max_dec_frame_reordering.
- Eliminated unused DIV operator definition and clarified other operators
- Eliminated pre-roll count from random access point SEI message
- Clarified aspects of start-code-related things now that the new Bossen end-of-slice CABAC stuffing is in.

Modifications in JVT-E146d24 relative to JVT-E146d23

- Clarification of edges controlled by macroblocks in slice
- Clarification of terminology

Modifications in JVT-E146d25 relative to JVT-E146d24

- Added statement "A given reference picture may appear at more than one index position in the remapped list 0 or list 1" (should probably also mention this concept in weighted prediction section, and should probably disallow multiple entries for the same picture unless weighted prediction is in use)

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

- changed names of mb_types and mb prediction types. applied changes to all clauses except 9 ff:
  - Introduced I, SI, P, and B collective macroblock types
  - Introduced Intra_4x4 and Intra_16x16 prediction modes
  - Introduced Intra and Inter prediction modes als collective terms
  - We may be able to remove names for mb_type
- changed "Category" in tables to "C" to save space
- changed mb prediction type indices from 1&2 to 0&1.
- changed syntax table index naming to MbPartIdx , SubMbIdx, SubMbPartIdx
- introduced column in I macroblock table for Intra_16x16 prediction mode showing Intra16x16PredMode, CodedBlockPatternChroma, CodedACblockPatternLuma
- improved clause 9.3
- minor changest to clause 0 (introduction)
- minor changes to clause 3 (definitions).
- minor revisions of clause 4 (abbreviations) – removed SAR abbreviation (was not used in the text anywhere) and SMPTE abbreviation (only used in a few places in Annex E, so just spelled it out when used)
- minor revisions of clause 5 (conventions) – added definition of "+=" and " –=", etc.
- minor revisions of subclauses 7.2 (definition of functions, categories, and descriptors) and 7.3 (syntax) - replacing u(n) with f(n) for several fixed-value syntax elements, removal of undefined three_reserved_bits syntax element, definition of read_bits(n), correction of several descriptor definitions to use read_bits(n) rather than next_bits(n) (because next_bits(n) does not advance the bitstream pointer), and typos.
- moved statement about which deblocking edges are controlled by a macroblock to the deblocking filter process subclause rather than the semantics subclause.
- improved transform level parsing subclause in clause 7.

Modifications in JVT-E146d35 relative to JVT-E146d34

- Changes from Miska and Ragip clarifying especially
  - SP pictures
  - required_frame_num_update_behaviour_flag
  - slice group stuff
- Changes from Gary centered on
  - inverse transform section
  - cross-references, typos, consistency issues, application of "keep with next" to prevent tables from splitting across page boundaries.
- [I suggest changing
  NumBitsInMbAddress → NumBitsInMBaddress,
  ChromaDCLevel → ChromaDClevel,
  ChromaACLevel → ChromaAClevel,
  MbIdx → MBidx,
  CtxMbTypeI → CtxMBtypeI,
  PicWidthInMbs → PicWidthInMBs
  etc.
  (the rule being that there is a case change when a new word starts – this way "MB" is macroblock instead of "Mb" and other acronyms stay as all-text]
- Replaced text marked in Portuguese (Brazil) with English (U.K.)

Modifications in JVT-E146d36 relative to JVT-E146d35

- CABAC clarifications
- Fix of num_ref_idx_active_l0_minus1 and num_ref_idx_active_l0_minus1 semantics for slices and bounds on values.
- Clarified next_bits(*n*) return value equal to 0 for insufficient data.
- Picture order count clarifications
- Removal of discussion of time in semantics of non_ref_pic_reset_flag

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

- Modification for consistency of use of terms luma and chroma and clarity of their definition
- Work on definitions and clarifications for MB -AFF
- Fix-up of slice_data() syntax bugs (elimination of first_nonskip_mb_in_pair(), provision for FMO, etc.)

Modifications in JVT-E146d37 relative to JVT-E146d36

- Changes from Detlev for further CABAC clarifications
- Changed slice termination syntax table for CABAC to correctly reflect JVT -E086 changes
- Replaced some (but perhaps not all) text labeled as "English (U.S.)" with "English (U.K.)"
- Replaced some text labeled as "German (Germany)" with "English (U.K.)"
- Replaced text marked in Portuguese (Brazil) with English (U.K.)
- Renamed syntax element names xyz_picture_xyz → xyz_pic_xyz and reference_xyz → ref_xyz and xyz_count_xyz → xyz_cnt_xyz for syntax naming consistency
- Fixed indentation in pic_timing( ) SEI message syntax
- Changed description of range of pan_scan_rect_id
- Changed description of checking for emulation prevention in NAL_unit( ) syntax
- Changed description of detection of end of byte streams.
- Clarified note describing potential external use of forbidden_bit.
- Clarified 172 fps limit in regard to accounting for field pictures
- Corrected calculation of AbsFrameNum with num_ref_frames_ in_pic_order_cnt_cycle equal to 0
- Clarified/corrected deblocking filter process from Anthony Joch
- Clarified/corrected parts of MB-AFF description
- Specified limits on range of a few syntax elements

Reminder: Please start all editorial notes with the string "[Ed".

Note: Anyone proposing any change to this draft must familiarize themselves with the content of the H.264/AVC drafting guide (version 7 or later) and the common text template. If you don't have these or don't know exactly what these are, ask Thomas or Gary for a copy.

End Ed. Notes]

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Title page to be provided by ITU-T | ISO/IEC

DRAFT INTERNATIONAL STANDARD
DRAFT ISO/IEC 14496-10 : 2002 (E)
DRAFT ITU-T Rec. H.264 (2002 E)
DRAFT ITU-T RECOMMENDATION

TABLE OF CONTENTS

Foreword xxiii
0 Introduction xxiv
0.1 Prolog xxiv
0.2 Purpose xxiv
0.3 Applications xxiv
0.4 Profiles and levels xxiv
0.5 Overview of the syntax xxv
0.5.1 Predictive coding xxv
0.5.2 Coding interlaced video xxv
0.5.3 Picture partition into macroblocks and smaller partitions xxv
0.5.4 Spatial redundancy reduction xxv
0.6 How to read this specification xxvi
1 Scope 1
2 Normative references 1
3 Definitions 1
4 Abbreviations 7
5 Conventions 8
5.1 Arithmetic operators 8
5.2 Logical operators 8
5.3 Relational operators 8
5.4 Bit-wise operators 8
5.5 Assignment 9
5.6 Mathematical functions 9
6 Source, coded, decoded, and output data formats 10
6.1 Bitstream formats 10
6.2 Source, decoded, and output picture formats 10
6.3 Spatial subdivision of pictures, slices, macroblocks, and sub -macroblocks 12
6.4 Inverse scanning processes 14
6.4.1 Inverse macroblock scanning process 14
6.4.2 Inverse macroblock pair scanning process 14
6.4.3 Inverse macroblock in macroblock pair scanning process 15
6.4.4 Inverse macroblock partition scanning process [Ed.Note: may not be needed as separate scan spec] 15
6.4.5 Inverse sub-macroblock partition scanning process [Ed.Note: may not be needed as separate scan spec] 15
6.4.6 Inverse 4x4 luma block scanning proces s 15
6.4.7 Derivation process of the availability for neighbouring macroblock addresses 15
6.4.8 Derivation process of the availabil ity for neighbouring macroblock pair addresses 16
7 Syntax and semantics 16
7.1 Method of describing the syntax in tabular form 16
7.2 Definitions of functions, categories and descriptors 17
7.3 Syntax in tabular form 19
7.3.1 NAL unit syntax 19
7.3.2 Raw byte sequence payloads and RBSP trailing bits syntax 20
7.3.2.1 Sequence parameter set RBSP syntax 20
7.3.2.2 Picture parameter set RBSP syntax 21
7.3.2.3 Supplemental enhancement information RBSP syntax 22
7.3.2.3.1 Supplemental enhancement information message syntax 22
7.3.2.4 Picture delimiter RBSP syntax 22

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

7.3.2.5 Filler data RBSP syntax 22
7.3.2.6 Slice layer RBSP syntax 23
7.3.2.7 Slice data partition RBSP syntax 23
7.3.2.7.1 Slice data partition A RBSP syntax 23
7.3.2.7.2 Slice data partition B RBSP syntax 23
7.3.2.7.3 Slice data partition C RBSP syntax 23
7.3.2.8 RBSP slice trailing bits syntax 24
7.3.2.9 RBSP trailing bits syntax 24
7.3.3 Slice header syntax 25
7.3.3.1 Reference index reordering syntax 26
7.3.3.2 Prediction weight table syntax 27
7.3.3.3 Decoded picture buffer management syntax 28
7.3.4 Slice data syntax 29
7.3.5 Macroblock layer syntax 30
7.3.5.1 Macroblock prediction syntax 31
7.3.5.2 Sub-macroblock prediction syntax 32
7.3.5.3 Residual data syntax 33
7.3.5.3.1 Residual 4x4 block CAVLC syntax 34
7.3.5.3.2 Residual 4x4 block CABAC syntax 35
7.4 Semantics *35*
7.4.1 NAL unit semantics 35
7.4.1.1 Example encoder procedure (informative) 37
7.4.2 Raw byte sequence payloads and RB SP trailing bits semantics 38
7.4.2.1 Sequence parameter set RBSP semantics 38
7.4.2.2 Picture parameter set RBSP semantics 39
7.4.2.3 Supplemental enhancement information RBSP semantics 42
7.4.2.3.1 Supplemental enhancement information message semantics 42
7.4.2.4 Picture delimiter RBSP semantics 42
7.4.2.5 Filler data RBSP semantics 42
7.4.2.6 Slice layer RBSP semantics 42
7.4.2.7 Slice data partition RBSP semantics 42
7.4.2.7.1 Slice data partition A RBSP semantics 42
7.4.2.7.2 Slice data partition B RBSP semantics 43
7.4.2.7.3 Slice data partition C RBSP semantics 43
7.4.2.8 RBSP slice trailing bits semantics 43
7.4.2.9 RBSP trailing bits semantics 43
7.4.3 Slice header semantics 43
7.4.3.1 Reference index reordering semantics 46
7.4.3.2 Prediction weight table semantics 47
7.4.3.3 Decoded picture buffer management semantics 47
7.4.4 Slice data semantics 49
7.4.5 Macroblock layer semantics 49
7.4.5.1 Macroblock prediction semantics 55
7.4.5.2 Sub-macroblock prediction semantics 56
7.4.5.3 Residual data semantics 57
7.4.5.3.1 Residual 4x4 block CAVLC semantics 58
7.4.5.3.2 Residual 4x4 block CABAC semantics 58
8 Decoding process 58
8.1 NAL unit decoding process *59*
8.2 Slice decoding process *59*
8.2.1 Detection process of coded picture boundaries 59
8.2.2 Decoding process for picture order count 59
8.2.2.1 Decoding process for picture order count type 0 60
8.2.2.2 Decoding process for picture order count type 1 61
8.2.2.3 Examples for the decoding process for picture order count types 0 and 1 (informative) 62
8.2.3 Decoding process for redundant slices 64
8.2.4 Decoding process for macroblock to slice group map 64
8.2.4.1 Specification of allocation order for box-out for decoding process for macroblock to slice group map 65
8.2.4.2 Specification of the mapping for raster scan for decoding process for macroblock to slice group map . 66
8.2.4.3 Specification of allocation order for wipe for decoding process for macroblock to slice group map 66
8.2.4.4 Specification of allocation order for macroblock level adaptive frame and field coding for decoding process for macroblock to slice group map 66
8.2.5 Decoding process for slice slice data partitioning 67

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

8.2.6 Decoding process for reference picture reordering 67
8.2.6.1 Decoding process for picture numbers 67
8.2.6.2 Decoding process for frame numbers 68
8.2.6.3 Initialization process for reference indices 69
8.2.6.3.1 Initialization process for reference indices for P and SP slices in frames 69
8.2.6.3.2 Initialization process for reference indices for P and SP slices in fields 69
8.2.6.3.3 Initialization process for reference indices for B slices in frames 70
8.2.6.3.4 Initialization process for reference indices for B slices in fields 71
8.2.6.4 Assignment process for reference indices 72
8.2.6.4.1 Assignment process for reference indices for short-term pictures 72
8.2.6.4.2 Assignment process for reference indices for long-term pictures 73
8.2.7 Decoded picture buffer management process 73
8.2.7.1 Sequence of operations for decoded picture buffer management process 74
8.2.7.2 Sliding window decoded picture buffer management process 74
8.2.7.3 Adaptive Memory Control decoded picture buffer management process 75
8.2.7.3.1 Assignment process for short-term pictures to "unused for reference" 75
8.2.7.3.2 Assignment process for long-term pictures to "unused for reference" 75
8.2.7.3.3 Assignment process for a long-term index to a short-term reference picture 75
8.2.7.3.4 Decoding process for maximum long-term index 76
8.2.7.3.5 Assignment process for all reference pictures to "unused for reference" 76
8.2.7.3.6 Assignment process for all non-reference pictures to "unused" 76
8.2.7.3.7 Assignment process for all reference and non-reference pictures to "unused" 77
8.2.8 Decoding process for macroblock level frame/field adaptive coding 77
8.2.8.1 Derivation process for frame or field decoding mode for a skipped macroblock pair in P or SP slices 78
8.2.8.2 Derivation process for frame or field decoding mode for skipped macroblock pair in B slice 78
8.3 Intra prediction process 78
8.3.1 Intra_4x4 prediction process 78
8.3.1.1 Derivation process for the available samples 78
8.3.1.1.1 Derivation process for the available samples for macroblocks 78
8.3.1.1.2 Derivation process for the available samples for macroblock pairs 79
8.3.1.2 Derivation process for the Intra4x4LumaPredMode 80
8.3.1.2.1 Derivation process for neighbouring macroblock addresses and their availability 81
8.3.1.2.2 Derivation process for neighbouring macroblock pair indices and their availability 81
8.3.1.2.3 Derivation process for the Intra_4x4 prediction modes from neighboring macroblocks 81
8.3.1.2.3.1 Derivation process for the Intra_4x4 prediction modes from left neighbouring macroblock 81
8.3.1.2.3.2 Derivation process for the Intra_4x4 prediction modes from above neighbouring macroblock 81
8.3.1.2.4 Derivation process for the Intra_4x4 prediction modes from neighboring macroblock pairs 82
8.3.1.2.4.1 Derivation process for the Intra_4x4 prediction modes from neighboring macroblock pairs when current macroblock pair is coded in frame decoding mode 82
8.3.1.2.4.2 Derivation process for the Intra_4x4 prediction modes from neighboring macroblock pairs when current macroblock pair is coded in field decoding mode 82
8.3.1.2.5 Derivation process for Intra4x4LumaPredMode for the current macroblock 84
8.3.1.2.6 Derivation process for Intra4x4LumaPredMode for the current macroblock pair 85
8.3.1.3 Decoding process for Intra_4x4 prediction modes 85
8.3.1.3.1 Specification of Intra_4x4_Vertical prediction mode 85
8.3.1.3.2 Specification of Intra_4x4_Horizontal prediction mode 85
8.3.1.3.3 Specification of Intra_4x4_DC prediction mode 86
8.3.1.3.4 Specification of Intra_4x4_Diagonal_Down_Left prediction mode 86
8.3.1.3.5 Specification of Intra_4x4_Diagonal_Down_Right prediction mode 86
8.3.1.3.6 Specification of Intra_4x4_Vertical_Right prediction mode 86
8.3.1.3.7 Specification of Intra_4x4_Horizontal_Down prediction mode 87
8.3.1.3.8 Specification of Intra_4x4_Vertical_Left prediction mode 87
8.3.1.3.9 Specification of Intra_4x4_Horizontal_Up prediction mode 87
8.3.2 Intra_16x16 prediction process 88
8.3.2.1 Mode 0: vertical prediction process 88
8.3.2.2 Mode 1: horizontal prediction process 88
8.3.2.3 Mode 2: DC prediction process 88
8.3.2.4 Mode 3: plane prediction process 89
8.3.3 Intra prediction process for chroma blocks 89
8.3.3.1 Mode 0: DC prediction process 89
8.3.3.2 Mode 1: horizontal prediction process 90
8.3.3.3 Mode 2: vertical prediction process 90
8.3.3.4 Mode 3: plane prediction 90

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

8.4 Inter prediction process *90*
8.4.1 Derivation process for motion vector components and reference pictur e indices 91
8.4.1.1 Derivation process for the neighbouring blocks for motion vector and reference index prediction 92
8.4.1.2 Derivation process for the luma motion vector predictors in the neighbouring blocks 93
8.4.1.2.1 Derivation process for the luma motion vector predictors in the neighbouring blocks in P and SP slices 93
8.4.1.2.2 Derivation process for the luma motion vector predictors in the neighbouring blocks in B slices .. 94
8.4.1.3 Derivation process for the luma motion vector prediction method 95
8.4.1.3.1 Decoding process for median luma motion vector prediction 95
8.4.1.3.2 Decoding process for directional segmentation luma motion vector prediction 95
8.4.1.4 Derivation process for luma motion vectors for skipped and Direct prediction macroblocks 96
8.4.1.4.1 Derivation process for luma motion vectors for skipped macroblocks in P and SP slices 96
8.4.1.4.2 Derivation process for luma moti on vectors for skipped, Direct_16x16 and Direct_8x8 in B slices 96
8.4.1.4.2.1 Derivation process for the co-located macroblock 97
8.4.1.4.2.2 Derivation process for spatial direct luma motion vector and reference picture index prediction mode 97
8.4.1.4.2.3 Derivation process for temporal direct luma motion vector and refer ence picture index prediction mode 98
8.4.1.5 Calculation of the resulting luma motion vector 102
8.4.1.6 [Ed, Note: This whole para graph is covered in 8.4.1]Derivation process for chroma motion vectors . 103
8.4.2 Decoding process for inter prediction samples 103
8.4.2.1 Reference picture selection process 104
8.4.2.2 Fractional sample interpolation process 104
8.4.2.2.1 Luma sample interpolati on process 104
8.4.2.2.2 Chroma sample interpolation process 106
8.4.2.3 Weighted sample prediction process 106
8.4.2.3.1 Default weighted sample prediction process 106
8.4.2.3.2 Weighted sample prediction process 107
8.5 Transform coefficient decoding process and picture construction process prior to deblocking filter process 109
8.5.1 Zig-zag scanning process and field scanning process 109
8.5.2 Scaling and transformation process 109
8.5.2.1 Scaling and transformation process for luma DC coefficients for Intra_16x16 macr oblock type 110
8.5.2.2 Scaling and transformation process for chroma DC coefficients 111
8.5.2.3 Scaling and transformation process for residual 4x4 blocks 111
8.5.3 Picture construction process prior to deblocking filter process 113
8.6 Decoding process for SP a nd SI slices 113
8.6.1 SP decoding process for non-switching pictures 113
8.6.1.1 Luma transform coefficient decoding process 114
8.6.1.2 Chroma transform coefficient decoding process 115
8.6.2 SP and SI slice decoding process for switching pictures 116
8.6.2.1 Luma transform coefficient decoding process 116
8.6.2.2 Chroma transform coefficient decoding process 117
8.7 Deblocking filter process 117
8.7.1 Derivation process for the content dependent boundary filtering strength 118
8.7.2 Derivation process for the thresholds for each block boundary 120
8.7.3 Filtering process for edges with Bs smaller than 4 121
8.7.4 Filtering process for edges for Bs equal to 4 122
9 Parsing process 123
9.1 Parsing process for exp-Golomb codes 123
9.1.1 Mapping process for unsigned exp -Golomb and for truncated unsigned exp -Golomb codes 124
9.1.2 Mapping process for signed Exp-Golomb codes 125
9.2 CAVLC parsing process for transform coefficients 125
9.2.1 Parsing process for total number of coeffi cients and trailing ones 125
9.2.1.1 Table selection process 128
9.2.2 Parsing process for level information 129
9.2.2.1 Table selection 132
9.2.3 Parsing process for run information 132
9.2.3.1 Parsing process for total number of zeros 132
9.2.3.2 Parsing process for run before each coefficient 134
9.3 CABAC parsing process for sli ce data 135
9.3.1 Initialisation process 135
9.3.1.1 Initialisation process for contexts 135

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

9.3.1.2 Initialisation process for the arithmetic decoding engine 145
9.3.1.3 Initialisation process for the arithmetic encoding engine (informative) 145
9.3.2 Binarization process 146
9.3.2.1 Unary (U) binarization process 147
9.3.2.2 Truncated unary (TU) binarization process 148
9.3.2.3 Concatenated unary/ k^{th}-order Exp-Golomb (UEGk) binarization process 148
9.3.2.4 Fixed-length (FL) binarization process 148
9.3.2.5 Binarization process for macroblock type and sub -macroblock type 149
9.3.2.6 Binarization process for motion vectors 152
9.3.2.7 Binarization process for coded block pattern 152
9.3.2.8 Binarization process for mb_qp_delta 152
9.3.3 Decoding process flow 153
9.3.3.1 Derivation process for the Ctxldx 153
9.3.3.1.1 Assignment process of CtxIdxInc for context templates using two neighbouring syntax elements 156
9.3.3.1.2 Assignment process of CtxIdxInc for context templates usi ng prior decoded bin values 158
9.3.3.1.3 Assignment process of CtxIdxInc for context identifiers used for transform coefficients 158
9.3.3.2 Arithmetic decoding process 159
9.3.3.2.1 Probability estimation process 160
9.3.3.2.2 Table-based arithmetic decoding pro cess 162
9.3.3.2.2.1 Decoding process for a decision 162
9.3.3.2.2.2 Renormalization process in the arithmetic decoding engine 163
9.3.3.2.2.3 Arithmetic decoding process for decisions with uniform pdf using arithmetic decoder bypass 164
9.3.3.2.2.4 Decoding process for decisions before termination 164
9.3.3.2.3 Table-based arithmetic encoding process (informative) 165
9.3.3.2.3.1 Encoding pro cess for a decision (informative) 166
9.3.3.2.3.2 Renormalization process in the arithmetic encoding engine (informative) 167
9.3.3.2.3.3 Writing process for compressed bytes (informative) 168
9.3.3.2.3.4 Arithmetic encoding process for decisions with uniform pdf using encoder bypass (informative) 169
9.3.3.2.3.5 Encoding process for a decision before termination (informative) 169
Annex A Profiles and levels 170
A.1 Requirements on video decoder capability 171
A.2 Profiles 171
A.2.1 Baseline profile 171
A.2.2 Main profile 171
A.2.3 X profile 172
A.3 Levels 172
A.3.1 Level limits 173
A.3.2 Decoded picture buffer capacity 173
A.3.3 Effect of level limits on frame rate (informative) 174
Annex B Byte stream format 174
B.1 Byte stream NAL unit syntax and semantics 175
B.1.1 Byte stream NAL unit syntax 175
B.1.2 Byte stream NAL unit semantics 175
B.2 Decoder byte-alignment recovery (informative) 175
Annex C Hypothetical Reference Decoder 176
C.1 Hypothetical reference decoder and buffering verifiers 176
C.1.1 Operation of coded picture buffer 177
C.1.1.1 Timing of bitstream arrival 177
C.1.1.2 Timing of coded picture removal 178
C.1.1.3 Conformance constraints on coded bitstreams 178
C.1.2 Operation of the decoded picture buffer (DPB) verifier 179
C.1.2.1 Removal of pictures from the DPB 179
C.1.2.2 Picture decoding 179
C.1.2.2.1 Storage of Reference Pictures into the DPB 179
C.1.2.2.2 Storage of Non-reference Pictures into the DPB 180
C.1.2.3 Unused picture marking 180
C.1.2.4 Conformance constraints 180
C.2 Informative description of the HRD 180
C.2.1 Constrained arrival time leaky bucket (CAT -LB) model 181

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

C.2.1.1 Operation of the CAT-LB HRD 181
C.2.1.2 Low-delay operation 184
C.2.1.3 Bitstream / packet stream constraints 184
C.2.1.3.1 Underflow 185
C.2.1.3.2 Overflow 185
C.2.1.3.3 Constant bitrate (CBR) operation 185
C.2.1.4 Rate control considerations 185
C.2.2 Multiple leaky bucket description 186
C.2.2.1 Schedule of a bitstream 186
C.2.2.2 Containment in a leaky bucket 186
C.2.2.3 Minimum buffer size and minimum peak rate 186
C.2.2.4 Encoder considerations 188

Annex D Supplemental enhancement information 188
D.1 SEI payload syntax 189
D.1.1 Buffering period SEI message syntax 190
D.1.2 Picture timing SEI message syntax 190
D.1.3 Pan-scan rectangle SEI message syntax 191
D.1.4 Filler payload SEI message syntax 191
D.1.5 User data registered by ITU-T Recommendation T.35 SEI message syntax 191
D.1.6 User data unregistered SEI message syntax 192
D.1.7 Random access point SEI message syntax 192
D.1.8 Decoded picture buffer management Repetition SEI message syntax 192
D.1.9 Spare picture SEI message syntax 193
D.1.10 Scene information SEI message syntax 193
D.1.11 Sub-sequence information SEI message syntax 193
D.1.12 Sub-sequence layer characteristics SEI message syntax 194
D.1.13 Sub-sequence characteristics SEI message syntax 194
D.1.14 Full-frame freeze SEI message syntax 194
D.1.15 Full-frame freeze release SEI message syntax 194
D.1.16 Full-frame snapshot SEI message syntax 195
D.1.17 Progressive refinement segment start SEI message syntax 195
D.1.18 Progressive refinement segment end SEI message syntax 195
D.1.19 Reserved SEI message syntax 195
D.2 SEI payload semantics 195
D.2.1 Buffering period SEI message semantics 195
D.2.2 Picture timing SEI message semantics 196
D.2.3 Pan-scan rectangle SEI message semantics 198
D.2.4 Filler payload SEI message semantics 199
D.2.5 User data registered by ITU-T Recommendation T.35 SEI message semantics 199
D.2.6 User data unregistered SEI message semantics 199
D.2.7 Random access point SEI message semantics 199
D.2.8 Decoded picture buffer management Repetition SEI message semantics 200
D.2.9 Spare picture SEI message semantics 200
D.2.10 Scene information SEI message semantics 201
D.2.11 Sub-sequence information SEI message semantics 202
D.2.12 Sub-sequence layer characteristics SEI message semantics 202
D.2.13 Sub-sequence characteristics SEI message semantics 203
D.2.14 Full-frame freeze SEI message semantics 204
D.2.15 Full-frame freeze release SEI message semantics 204
D.2.16 Full-frame snapshot SEI message semantics 204
D.2.17 Progressive refinement segment start SEI message semantics 204
D.2.18 Progressive refinement segment end SEI message semantics 205
D.2.19 Reserved SEI message semantics 205

Annex E Video usability information 206
E.1 VUI syntax 207
E.1.1 VUI sequence parameters syntax 207
E.1.2 HRD parameters syntax 208
E.1.3 VUI picture parameters syntax 208
E.2 VUI semantics 208
E.2.1 VUI sequence parameters semantics 208
E.2.2 HRD parameters semantics 216
E.2.3 VUI picture parameters semantics 217

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

LIST OF FIGURES

Figure 6-1 – Nominal vertical and horizontal locati ons of 4:2:0 luma and chroma samples in a frame 11

Figure 6-2 – Nominal vertical and temporal sampling locations of samples in 4:2:0 interlaced frames 12

Figure 6-3 – A picture with 11 by 9 macroblocks (QCIF picture) 13

Figure 6-4 – Partitioning of the decoded frame into macroblock pairs. 13

Figure 6-5 – Macroblock partitions, sub-macroblock partitions, inverse macroblock partition scans, and inverse sub-macroblock partition scans. [Ed. Note: Figure does not print properly on at least one printer.] 14

Figure 8-1 – Default reference field number assignment when the current picture is the first field coded in a frame [Ed. Note: Need to change f1/f2 to f0/f1 – always use zero-based indexing] 70

Figure 8-2 – Default reference field number assignment when the current picture is the second field coded in a frame [Ed. Note: Need to change f1/f2 to f0/f1 – always use zero-based indexing] 70

Figure 8-3 – Split of a pair of macroblocks into one top-field macroblock and one bottom-field macroblock. [Ed. Note: Should hae 16 lines per MB, not 14 as shown above] 77

Figure 8-4 – Labelling of samples used for Intra_4x4 prediction 79

Figure 8-5 – Intra_4x4 prediction mode directions (informative) 80

Figure 8-6 a) Definition of above neighbouring blocks when the above macroblock pair is in field mode b) Definition of above neighbouring blocks when the above macroblock pair is in frame mode c) Definition of left neighbouri ng blocks when the left macroblock pair is in field mode d) Definition of left neighbouring blocks when the left macroblock pair is in frame mode (informative) 84

Figure 8-7 – Determination of the neighbouring luma blocks and associated motion vectors 92

Figure 8-8 – Differential motion vector decoding with scaled motion vector 95

Figure 8-9 – Directional segmentation prediction 96

Figure 8-10 – Double-dashed border to indicate each 'corner' 4x4 block of each co-located 8x8 submacroblock, from which each temporal direct-mode 8x8 sub-macroblock derives its motion vectors if direct_temporal_constrained_flag is set to 1. 98

Figure 8-11 – Both the current block and its co-located block in the list 1 reference picture are in frame mode (f0 and f1 indicate the corresponding fields) 100

Figure 8-12 – Both the current macroblock and its co-located macroblock in the temporally subsequent picture are in field mode. 100

Figure 8-13 – The list 0 motion vector of the co-located block in field 1 of the list 1 reference frame may point to field 0 of the same frame. 101

Figure 8-14 – The current macroblock is in field mode and its co-located macroblock in the list 1 reference picture is in frame mode 101

Figure 8-15 – The current macroblock is in frame mode while its co-located macroblock in the list 1 reference picture is in field mode. 102

Figure 8-16 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation. 105

Figure 8-17 – Fractional sample position dependent variables in chroma interpolation a nd surrounding integer position samples A, B, C, and D. [Ed. Note: Edit to replace s with 8 and d^x and d^y with d_x and d_y] 106

Figure 8-18 – a) Zig-zag scan. b) Field scan 109

Figure 8-19 – A block diagram of a conceptual decoder for non-intra coded macroblocks in SP slices in which sp_for_switch_flag is equal to 0. 114

Figure 8-20– A block diagram of a conceptual decoder for non-intra macroblocks in SI slices; and for non-intra coded macroblocks in SP slices in which sp_for_switch_flag is equal to 1. (informative) 116

Figure 8-21 – Boundaries in a macroblock to be filtered (luma boundaries shown with solid lines and chroma boundaries shown with dashed lines) 118

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Figure 8-22 – Flow chart for determining the boundary strength (Bs), for the block boundary between two neighbouring 4x4 luma blocks [Ed.Note: ?] p and q, where $V_i(p,x)$ and $V_i(p, y)$ for i=1 to 1, 2 or 4, are the horizontal and vertical components of the motion vectors of block [Ed.Note: ?] p for th e reference frame or the first and second reference fields. 120

Figure 8-23 – Convention for describing samples across a 4x4 block horizontal or vertical boundary 120

Figure 9-1 – Flowchart of initialisation of the decoding engine 145

Figure 9-2 – Flowchart of initialisation of the encoding engine 146

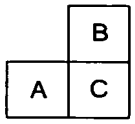


Figure 9-3 – Illustration of the generic context template using two neighbouring syntax elements A and B for conditional decoding of a current syntax element C 156

Figure 9-4 – Overview of the arithmetic decoding process (informative) [Ed.Note: Either Flowchart has to be updated or removed] 162

Figure 9-5 – Flowchart for decoding a decision 163

Figure 9-6 – Flowchart of renormalization 164

Figure 9-7 – Flowchart of decoding bypass 164

Figure 9-8 – Flowchart of decoding a decision before termination 165

Figure 9-9 – Overview of the arithmetic encoding process 166

Figure 9-10 – Flowchart for encoding a decision 167

Figure 9-11 – Flowchart of renormalization 168

Figure 9-12 – Flowchart of putting bit 168

Figure 9-13 – Flowchart for output of compressed bytes 169

Figure 9-14 – Flowchart of encoding bypass 169

Figure 9-15 – Flowchart of encoding a decision before termination 170

Figure C-1 – Structure of Byte streams and NAL unit streams and HRD Conformance Points 176

Figure C-2 – HRD Buffer Verifiers 177

Figure C-3 – A Hypothetical Reference Decoder 180

Figure C-4 – Buffer fullness plot for example HRD in Table C-2 with picture sizes given in Table C-3 183

Figure C-5 – Illustration of the leaky bucket concept [Ed. Note: Something wrong with figure] 186

Figure C-6 – Further illustration of the leaky bucket concept 187

Figure E-1 – Luma and chroma sample types for frames 214

Figure E-2 – Luma and chroma association for fields 214

LIST OF TABLES

Table 7-1 – NAL unit type codes 36

Table 7-2– Refined macroblock to slice group map type 41

Table 7-3– Meaning of pic_type 42

Table 7-4 – Name association to slice_type 44

Table 7-5 – remapping_of_pic_nums_idc operations f or re-mapping of reference pictures 46

Table 7-6 – Interpretation of adaptive_ref_pic_buffering_mode_flag 48

Table 7-7 – Memory management control operation (memory_management_control_operation) values 49

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 7-8 – Allowed collective macroblock types for slice_type 50
Table 7-9 – Macroblock types for I slices 50
Table 7-10 – Macroblock type with value 0 for SI slices 51
Table 7-11 – Macroblock type values 0 to 4 for P and SP slices 51
Table 7-12 – Macroblock type values 0 to 22 for B slices 52
Table 7-13 – Specification of CodedBlockPatternChroma values 53
Table 7-14 – Assignment of coded_block_pattern codewords for macroblock prediction types 54
Table 7-15 – Relationship between intra_chroma_pred_mode and spatial prediction modes 55
Table 7-16 – Reference frame indicated by ref_idx_10 for frame picture (informative) 56
Table 7-17 – Reference field indicated by ref_idx_10 for field picture (informative) 56
Table 7-18 – Sub-macroblock types in P macroblocks 57
Table 7-19 – Sub-macroblock types in B macroblocks 57
Table 8-1 – First example for picture order count type 0 [Ed. Note: Editorial issue with first column – Do we have frame types? What purpose does that column if any?] 62
Table 8-2 – Second example for picture order count type 0 63
Table 8-3 – Third example for picture order count type 0 [Ed. Note: Consistency – Should this have a frame_num column?] 63
Table 8-4 – First example for picture order count type 1 64
Table 8-5 – Second example for picture order count type 1 64
Table 8-6 – Mapping for the box-out macroblock map to slice group type 66
Table 8-7 – Specification of Intra4x4LumaPredMode[4x4LumaBlkIdx] and associated names 80
Table 8-8 – Calculation of vertical component of chroma vector in field coding mode 103
Table 8-9 – Specification of QP_C as a function of QP_I 110
Table 8-10 – QP_{av} and offset dependent threshold parameters α and β 121
Table 8-11 – Value of filter clipping parameter C0 as a function of $Index_A$ and Bs 122
Table 9-1 – Bit strings with "prefix" and "suffix" bits and assignment to CodeNum ranges 124
Table 9-2 – Code number and Exp-Golomb codewords in explicit form and used as ue(v) 124
Table 9-3 – Assignment of syntax element to CodeNum for signed exp-Golomb coded syntax elements se(v) 125
Table 9-4 – coeff_token mapping to to tal_coeff( ) and trailing_ones( ): Num-VLC0 126
Table 9-5 – coeff_token mapping to total_coeff( ) and trailing_ones( ): Num-VLC1 127
Table 9-6 – coeff_token mapping to total_coeff( ) and trailing_ones( ): Num-VLC2 128
Table 9-7 – coeff_token mapping to total_coeff( ) and trailing_ones( ): Num-VLC_Chroma_DC 128
Table 9-8 – Calculation of N for Num-VLCN 129
Table 9-9 – Level tables 129
Table 9-10 – Level VLC1 130
Table 9-11 – Level VLC2 130
Table 9-12 – Level VLC3 131
Table 9-13 – Level VLC4 131
Table 9-14 – Level VLC5 131
Table 9-15 – Level VLC6 131
Table 9-16 – total_zeros tables for all 4x4 blocks 133

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-17 – TotalZeros table for chroma DC 2x2 blocks 134
Table 9-18 – Tables for run_before 134
Table 9-19 – Initialisation parameters for context identifiers CtxMbTypeI, CtxMbTypeSIPrefix, and CtxMbTypeSISuffix 136
Table 9-20 – Initialisation parameters for context identifier CtxMbSkip, CtxMbTypeP, CtxMbTypePSuffix, and CtxMbTypeBSuffix 136
Table 9-21 – Initialisation parameters for context identifier CtxMbTypeB, CtxB8ModeB, and CtxB8ModeP 136
Table 9-22 – Initialisation parameters for context identifiers CtxAbsMvdHoriz and CtxAbsMvdVert 137
Table 9-23 – Initialisation parameters for context identifier CtxRefIdx and CtxMbField 137
Table 9-24 – Initialisation parameters for context identifiers CtxMbQpDelta, CtxIpredChroma, CtxIpredLumaMpm, CtxIpredLumaRm 137
Table 9-25 – Initialisation parameters for context identifiers CtxCbpLuma and CtxCbpChroma 138
Table 9-26 – Initialisation parameters for context identifier CtxCodedBlock 139
Table 9-27 – Initialisation parameters for context identifier CtxSigCoeff 140
Table 9-28 – Initialisation parameters for context identifier CtxLastCoeff 142
Table 9-29 – Initialisation parameters for context identifier CtxAbsLevel 144
Table 9-30 – Syntax elements and associated context identifiers 147
Table 9-31 – Unary binarization (informative) 148
Table 9-32 – Binarization for macroblock types for I slices 150
Table 9-33 – Binarization for macroblock types for P, SP, and B slice s 151
Table 9-34 – Binarization for sub-macroblock types in P, SP, and B slices 152
Table 9-35 – Context identifiers and associated MaxBinCtxIdx and ranges of CtxIdx 154
Table 9-36 – Assignment of CtxIdxInc to BinIdx for all context identifiers except CtxCodedBlock, CtxSigCoeff, CtxLastCoeff, and CtxAbsLevel 155
Table 9-37 – Assignment of CtxIdxInc to CtxCat for context identifiers CtxCodedBlock, CtxSigCoeff, and CtxLastCoeff 155
Table 9-38 – Assignment of CtxAbsLevelOffset and CtxIdxInc to CtxCat for context identifier CtxAbsLevel 155
Table 9-39 – Specification of CondTerm for context identifiers and BinIdx using context templates according to Equations (9-2) – (9-4) 157
Table 9-40 – Specification of CondTerm for context identifiers and BinIdx using the context template according to Equation 9-6 158
Table 9-41 – Context categories for the different block types 159
Table 9-42 – Probability transition 160
Table 9-43 – RTAB[State][QuantCodeIntRange] table for interval subdivision 161
Table A-1 – Level Limits 173
Table A-2 – ChromaFormatParameter values 173
Table A-3 – Maximum frame rates for some common picture sizes (Hz) 174
Table C-1 - Attributes of an example CAT-LB HRD 181
Table C-2 - Picture sizes, and encoding, arrival and removal times for the example CAT -LB HRD 182
Table D-1 – Interpretation of pic_structure 196
Table D-2 – Mapping of ct_type to original picture scan 197
Table D-3 – Definition of counting_type values 197
Table D-4 – Scene transition types. 201

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table E-1 – Meaning of sample aspect ratio indicator 209
Table E-2 – Meaning of video_format 209
Table E-3 – Colour Primaries 211
Table E-4 – Transfer Characteristics 212
Table E-5 – Matrix Coefficients 213
Table E-6 – Chroma Sampling Structure for Frame Pictures 213
Table E-7 – Chroma Sampling Structure for Fields 214

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Foreword

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU -T) is a permanent organ of ITU. ITU -T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis. The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU -T study groups which, in turn, produce Recommendations on these topics. The approval of ITU -T Recommendations is covered by the procedure laid down in WTSA Resolution 1. In some areas of information technology which fall within ITU -T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

ISO (the International Organisation for Standardisation) and IEC (the International Electrotechnical Commission) form the specialised system for world-wide standardisation. National Bodies that are members of ISO and IEC participate in the development of International Standards throu gh technical committees established by the respective organisation to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organisations, governmental and non -governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC1. Draft International Standards adopted by the joint technical committee are ci rculated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

This Recommendation | International Standard is being submitted for approval to the ITU -T and ISO/IEC JTC1/SC29. It was prepared jointly by ITU-T SG16 Q.6 also known as VCEG (Video Coding Experts Group) and by ISO/IEC JTC1/SC29/WG11, also known as MPEG (Moving Picture Experts Group). VCEG was formed in 1997 to maintain prior ITU-T video coding standards an d develop new video coding standard(s) appropriate for a wide range of conversational and non-conversational services. MPEG was formed in 1988 to establish standards for coding of moving pictures and associated audio for various applications such as digita l storage media, distribution and communication.

In this Recommendation | International Standard Annexes A through E contain normative requirements and are an integral part of this Recommendation | International Standard.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

0 Introduction

0.1 Prolog

As processing power and memory costs have reduced, network support for coded video data has diversified, and advances in video coding technology have progressed, the need has arisen for an industy standard for compressed video representation with substantially increase d coding efficiency and enhanced robustness to network environments. Toward these ends the ITU-T video coding experts group (VCEG) and the ISO/IEC moving picture experts group (MPEG) formed a joint video team (JVT) in 2001 for development of a new ITU -T Recommendation | International Standard.

0.2 Purpose

This Recommendation | International Standard was developed in response to the growing need for higher compression of moving pictures for various applications such as videoconferencing, digital storage media, television broadcasting, internet streaming and communication. It is also designed to enable the use of the coded video representation in a flexible manner for a wide variety of network environments. The use of this Recommendation | Insternational Standar d allows motion video to be manipulated as a form of computer data and to be stored on various storage media, transmitted and received over existing and future networks and distributed on existing and future broadcasting channels.

0.3 Applications

This Recommendation | International Standard is designed to cover a broad range of applications for video content including but not limited to the following:

CATV	Cable TV on optical networks, copper, etc.
DBS	Direct broadcast satellite video services
DSL	Digital subscriber line video services
DTTB	Digital terrestrial television broadcasting
ISM	Interactive storage media (optical disks, etc.)
MMM	Multimedia mailing
MSPN	Multimedia services over packet networks
RTC	Real-time conversational services (videoconferencing, videophone, etc.)
RVS	Remote video surveillance
SSM	Serial storage media (digital VTR, etc.)

0.4 Profiles and levels

This Recommendation | International Standard is designed to be generic in the sense that it serves a wide range of applications, bit rates, resolutions, qualities and services. Applications should cover, among other things, digital storage media, television broadcasting and real-time communications. In the course of creating this Specification, various requirements from typical applications have been considered, necessary algorithmic elements have been developed, and they have been integrated into a single syntax. Hence, this Specification will facilitate video data interchange among different applications.

Considering the practicality of impleme nting the full syntax of this Specification, however, a limited number of subsets of the syntax are also stipulated by means of "profile" and "level". These and other related terms are formally defined in clause 4.

A "profile" is a subset of the entire bitstream syntax that is specified by this Recommendation | International Standard. Within the bounds imposed by the syntax of a given profile it is still possible to require a very large variation in the performance of encoders and decoders depending upon the values taken by parameters in the bitstream such as the specified size of the decoded pictures. It is currently neither practical nor economic to implement a decoder capable of dealing with all hypothetical uses of the syntax within a particular profile.

In order to deal with this problem, "levels" are specified within each profile. A level is a specified set of constraints imposed on parameters in the bitstream. These constraints may be simple limits on numbers. Alternatively they may take the form of constraints on arithmetic combinations of the parameters (e.g. frame width multiplied by frame height multiplied by frame rate).

Coded video content conforming to this Specification uses a common syntax. In order to achieve a subs et of the complete syntax, flags and parameters are included in the bitstream that signal the presence or otherwise of syntactic elements that occur later in the bitstream. In order to specify constraints on the syntax (and hence define a profile), it is

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

thus only necessary to constrain the values of these flags and parameters that specify the presence of later syntactic elements. [Ed. Note: This is not strictly true. There are characteristics like ASO that have no flag that control them yet are used to define profiles.]

0.5 Overview of the syntax

The coded representation specified in the syntax achieves a high compression capability for a desired image quality. The algorithm is not lossless as the exact source sample values are typically not preserved through the encoding and decoding processes. A number of techniques may be used to achieve highly efficient compression. The expected encoding algorithm (not specified in this Recommendation | International Standard) in inter coding first uses block-based inter prediction to exploit temporal statistical dependencies or in intra coding first uses spatial prediction to exploit spatial statistical dependencies in the source signal. Motion vectors and intra prediction modes may be specified for a variety of block sizes in the picture. The prediction residual is then further compressed using a transform to remove spatial correlation inside the transform block before it is quantised, producing an irreversible process that typically discards less important information while forming a close approximation to the source samples. Finally, the motion vectors or intra prediction modes are combined with the quantised transform coefficient information and encoded using either variable length codes or arithmetic coding.

0.5.1 Predictive coding

Because of the conflicting requirements of random access and highly efficient compression, two main coding types are specified. Intra coding is done without reference to other pictures. Intra coding may provide access points to the coded sequence where decoding can begin and continue correctly, but typically also shows only moderate compression efficiency. Inter coding (predictive or bi-predictive) is more efficient using inter prediction of each block of sample values from some previously decoded picture selected by the encoder. In contrast to some other video coding standards, pictures using bi-predictive inter coding may be used as references for inter coding of other pictures.

The application of the three coding types to pictures in a sequence is flexible, and the order of the decoding process is generally not the same as the order of the source picture capture process in the encoder or the output order from the decoder for display. The choice is left to the encoder and will depend on the requirements of the application. The decoding order is specified such that the decoding of pictures that use inter-picture prediction follows later in decoding order than other pictures that are referenced in the decoding process.

0.5.2 Coding interlaced video

Each frame of interlaced video consists of two fields which are separated in capture time. This Recommendation | International Standard allows either the representation of complete frames or the representation of individual fields. Frame encoding or field encoding can be adaptively selected on a picture-by-picture basis and also on a more localized basis within a coded frame. Frame encoding is typically preferred when the video scene contains significant detail with limited motion. Field encoding, in which the second field can be predicted from the first, works better when there is fast picture-to-picture motion.

0.5.3 Picture partition into macroblocks and smaller partitions

As in previous video coding Recommendations and International Standards, a macroblock consisting of a 16x16 block of luma samples and two corresponding blocks of chroma samples is used as the basic processing unit of the video decoding process.

A macroblock can be further partitioned for inter prediction. The selection of the size of inter prediction partitions is a result of a trade-off between the coding gain provided by using motion compensation with smaller blocks and the quantity of data needed to represent the data for motion compensation. In this Recommendation | International Standard the inter prediction process can form segmentations for motion representation as small as 4x4 luma samples in size, using motion vector accuracy of one-quarter of the luma sample grid spacing displacement. The inter prediction process for inter prediction of a sample block can also involve the selection of the picture to be used as the reference picture from a number of stored previously-decoded pictures. Motion vectors are encoded differentially with respect to predicted values formed from nearby encoded motion vectors.

Typically, the encoder calculates appropriate motion vectors and other data elements represented in the video data stream. This motion estimation process in the encoder and the selection of whether to use inter prediction for the representation of each region of the video content is not specified in this Recommendation | International Standard.

0.5.4 Spatial redundancy reduction

Both source pictures and prediction residuals have high spatial redundancy. This Recommendation | International Standard is based on the use of a block-based transform method for spatial redundancy removal. After inter prediction from previously-decoded samples in other pictures or spatial-based prediction from previously-decoded samples within the current picture, the resulting prediction residual is split into 4x4 blocks. These

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

are converted into the transform domain where they are quantised. After quantisation many of the transform coefficients are zero or have low amplitude and can thus be represented with a small amount of enco ded data. The processes of transformation and quantization in the encoder are not specified in this Recommendation | International Standard.

0.6 How to read this specification

It is recommended that the reader starts with clause 1 (Scope) and moves on to clause 3 (Definitions). Clause 6 should be read for the geometrical relationship of the source, input, and output of the decoder. Clause 7 (Syntax and semantics) specifies the order to parse syntax elements from the bitstream. See subclauses 7.1-7.3 for syntax order and see subclause 7.4 for semantics, i.e., the scope, restrictions, and conditions that are imposed on the syntax elements. The actual parsing for most syntax elements is specified in clause 9 (Parsing process). Finally, clause 8 (Decoding process) specifies how the syntax elements are mapped into decoded samples. Throughout reading this specification, the reader should refer to clauses 2 (Normative references), 4 (Abbreviations), and 5 (Conventions) as needed. [Ed. Note: And what about clause 9 and the five annexes?]

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

1 Scope

This document specifies ITU-T Recommendation H.264 | ISO/IEC International Standard ISO/IEC 14496-10 video coding.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

- ITU-T Recommendation T.35 (2000), *Procedure for the allocation of ITU-T defined codes for non-standard facilities*

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 **AC coefficient**: Any *transform coefficient* for which the frequency index in one or both dimensions is non-zero.

3.2 **arbitrary slice ordering**: An ordering of *slices* in which the *macroblock address* of the first *macroblock* of some *slice* may be smaller than the *macroblock address* of the first *macroblock* of some other *slice* of the *picture* that precedes it in *decoding order*.

3.3 **B slice**: A *bi-predictive slice*.

3.4 **bi-predictive slice**; A *slice* that is coded in a manner in which a weighted average of two *inter prediction blocks* may be used for *inter prediction*.

3.5 **bitstream**: A sequence of bits that forms the representation of data and coded *fields* and *frames*. Bitstream is a collective term used to refer either to the *NAL unit stream* or the *byte stream format*.

3.6 **block**: An MxN (M-column by N-row) array of samples, or an MxN array of *transform coefficients*.

3.7 **bottom field**: One of two *fields* that comprise a *frame*. Each row of a *bottom field* is spatially located immediately below a corresponding row of a *top field*.

3.8 **bottom macroblock (of a macroblock pair)**: The *macroblock* within a *macroblock pair* that contains the samples in the bottom row of samples for the *macroblock pair*. For a *field macroblock pair*, the bottom macroblock represents the samples from the region of the *bottom field* of the *frame* that lie within the spatial region of the *macroblock pair*. For a *frame macroblock pair*, the bottom macroblock represents the samples of the *frame* that lie within the bottom half of the spatial region of the *macroblock pair*.

3.9 **byte**: A sequence of 8 bits, ordered from the first and most significant bit on the left to the last and least significant bit on the right.

3.10 **byte-aligned**: A bit in a *bitstream* is *byte-aligned* if its position is a multiple of 8 bits from the first bit in the *bitstream*.

3.11 **byte stream format**: An encapuslation of a *NAL unit stream* containing *start code prefixes* and *NAL units* as specified in Annex B.

3.12 **category**: A number associated with each syntax element that specifies the allocation of syntax elements to *NAL units* for *slice data partitioning*. It may also be used by the systems *layer* to refer to classes of syntax elements in a manner not specified in this Recommendation | International Standard.

3.13 **chroma**: An adjective specifying that a sample array or single sample is representing one of the two colour difference signals related to the primary colours. The symbols used for a chroma array or sample are Cr and Cb. The term chroma is used rather than the term chrominance in order to avoid the implication of the use of linear light transfer characteristics that is often associated with the term chrominance.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

3.14 **coded field**: A coded representation of a *field*.

3.15 **coded frame**: A coded representation of a *frame*.

3.16 **coded picture buffer (CPB)**: A first-in first-out buffer containing coded pictures in *decoding order* specified in the *hypothetical reference decoder* in Annex C.

3.17 **coded representation**: A data element as represented in its coded form.

3.18 **common intermediate format (CIF)**: A video frame that is 22 *macroblocks* wide and 18 *macroblocks* high.

3.19 **component**: An array or single sample from one of the three arrays (*luma* and two *chroma*) that make up a *field* or *frame*.

3.20 **context**: The numerical value of the *context variable* when decoding a *symbol*.

3.21 **context modelling**: The choice and specification of prior *decoded symbols* that are to be used in the *decoding* of a *symbol*.

3.22 **context variable**: A parameter specified for the decoding process of a *symbol* by an equation containing recently *decoded symbols* as defined by *context modelling*.

3.23 **DC coefficient**: The *transform coefficient* for which the frequency index is zero in both dimensions. [Ed.Note: define frequency index]

3.24 **decoded picture**: A *decoded picture* is obtained by decoding a *coded picture*. A *decoded picture* is either a *decoded frame*, or a *decoded field*. A *decoded field* is either a *decoded top field* or a *decoded bottom field*.

3.25 **decoded picture buffer (DPB)**: A buffer holding *decoded pictures* for reference, output reordering, or output timing.

3.26 **decoded picture buffer management**: specifies in the coded data, how the *decoding process* applies to the *decoded pictures buffer* and in particular to the *decoded picture buffer*. [Ed.Note: revisit]

3.27 **decoder**: An embodiment of a *decoding process*.

3.28 **decoding order**: The nominal order in which syntax elements are processed by the *decoding process*.

3.29 **decoding process**: The process specified in this Recommendation | International Standard that reads a *NAL unit stream* and produces decoded *fields* or *frames*.

3.30 **direct prediction**: An *inter prediction* for a *block* for which no *motion vector* is decoded. [Ed.Note: Is There a need to define temporal and spatial]

3.31 **encoder**: An embodiment of an *encoding process*.

3.32 **emulation prevention byte**: A byte equal to 0x03 that may be present within a *NAL unit*. The presence of emulation prevention bytes ensures that no sequence of consecutive byte -aligned bytes in the *NAL unit* contains a *start code prefix*.

3.33 **encoding process**: A process, not specified in this Recommendation | International Standard, that reads a sequence of *fields* and *frames* and produces a conforming *NAL unit stream* as specified in this Recommendation | International Standard.

3.34 **field**: An assembly of alternate rows of a *frame*. A *frame* is composed of two *fields*, a *top field* and a *bottom field*.

3.35 **field macroblock**: A macroblock containing samples from a single *field*. All *macroblocks* of a *coded field* are field macroblocks. When *macroblock-adaptive frame/field decoding* is in use, some macroblocks of a *coded frame* may be field macroblocks.

3.36 **field macroblock pair**: A *macroblock pair* decoded as two *field macroblocks*.

3.37 **field scan**: A specific sequential ordering of *transform coefficients* that differs from the *zig-zag* scan by scanning columns more rapidly than rows. Field scan is used for *transform coefficients* in *field macroblocks*.

3.38 **flag**: A variable which can take one of only two possible values.

3.39 **frame**: A *frame* contains an array of luma samples and two corresponding arrays of chroma samples. A *frame* consists of two *fields*, a *top field* and a *bottom field*. For interlaced video, one of these *fields* is sampled temporally later than the other.

3.40 **frame macroblock**: A *macroblock* representing samples from two *fields* of a *coded frame*. When *macroblock-adaptive frame/field decoding* is not in use, all macroblocks of a *coded frame* are frame macroblocks. When

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

macroblock-adaptive frame/field decoding is in use, some macroblocks of a *coded frame* may be frame macroblocks.

3.41 **field macroblock pair**: A *macroblock pair* decoded as two *frame macroblocks*.

3.42 **frame number**: A number assigned to each video *frame* for use as a frame identifier in the *decoding process*.

3.43 **hypothetical reference decoder (HRD)**: A hypothetical *decoder* model that specifies a constraint on the variability of the *NAL unit stream* that an encoder or editing process may produce.

3.44 **I slice**: An *intra slice*.

3.45 **instantaneous decoder refresh (IDR) picture**: A *coded picture* containing only *I* or *SI slice types* that causes the *decoder* to mark all *reference pictures* in the *decoded picture buffer* as unused for reference immediately before *decoding* the *IDR picture*, and to indicate that later *coded pictures* in *decoding order* can be *decoded* without *inter prediction* from any *picture* decoded prior to the *IDR picture*.

3.46 **inter coding**: Coding of a *block*, *macroblock*, *slice*, or *picture* that uses *inter prediction*.

3.47 **inter prediction**: A *prediction* derived from decoded samples of *pictures* other than the current *decoded picture*. *Inter prediction* is a collective term for the *prediction process* of some *macroblocks* in *P*, *SP*, and *B slices*.

3.48 **interlaced video:** A form of video content in which each *frame* represents two *fields* which are separated in capture time.

3.49 **intra coding**: Coding of a *block*, *macroblock*, *slice* or *picture* that uses *intra prediction*.

3.50 **intra prediction**: A *prediction* derived from the decoded samples of the same *decoded picture*. [Ed. Note: collective term of ?]

3.51 **intra slice:** A *slice* that is coded using prediction only from decoded samples within the same *slice*.

3.52 **inverse transform**: A part of the *decoding process* by which a *block* of *transform coefficients* is converted into a *block* of spatial-domain values, or by which a *block* of *transform coefficients* is converted into a *block* of *DC coefficients*.

3.53 **layer**: One of a set of syntactical structures in a non-branching hierarchical relationship. Higher layers contain lower layers. The coding layers are the sequence, *picture*, *slice*, reference picture selection, *macroblock*, prediction block and 4x4 block layers.

3.54 **level**: A defined set of constraints on the values which may be taken by the parameters of this Recommendation | International Standard. The same set of level definitions are used with all profiles, but individual implementations may support a different level for each supported profile. In a different context, *level* is the value of a *transform coefficient* prior to *scaling*.

3.55 **luma**: An adjective specifying that a sample array or single sample is representing the monochrome signal related to the primary colours. The symbol used for luma is Y. The term luma is used rather than the term luminance in order to avoid the implication of the use of linear light transfer characteristics that is often associated with the term luminance.

3.56 **macroblock**: A 16x16 *block* of *luma* samples and two corresponding blocks of *chroma* samples. The division of a *slice* or a *macroblock pair* into macroblock is a *partitioning*.

3.57 **macroblock address**: When *macroblock-adaptive frame/field decoding* is not in use, a macroblock address is the *raster scan order* number of a macroblock starting with zero for the top left *macroblock* in a *picture*. When *macroblock-adaptive frame/field decoding* is in use, a macroblock address is 2 times the *macroblock pair address* for the *top macroblock*, and is two times the *macroblock pair address* plus 1 for the *bottom macroblock*.

3.58 **macroblock-adaptive frame/field decoding**: A *decoding process* for *coded frames* in which some *macroblocks* may be decoded as *frame macroblocks* and others may be decoded as *field macroblocks*.

3.59 **macroblock to slice group map**: A means of categorizing the *macroblocks* of a picture into *slice groups*. The macroblock to slice group map consists of an array of numbers, one for each coded *macroblock*, indicating the *slice group* to which each coded *macroblock* belongs.

3.60 **macroblock location**: The two dimensional coordinates of a *macroblock* in a *picture* denoted by (x,y). For the top left *macroblock* of the *picture* (x,y)=(0,0). x is incremented by 1 for each *macroblock* column from left to right. When *macroblock-adaptive frame/field decoding* is not in use, y is incremented by 1 for each *macroblock* row from top to bottom. When *macroblock-adaptive frame/field decoding* is in use, y is

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

incremented by 2 for each *macroblock pair* row from top to bottom, and is incremented by an additional 1 if a macroblock is a *bottom macroblock*.

3.61 **macroblock pair**: A pair of vertically-contiguous *macroblocks* in a *picture* that is coupled for use in *macroblock-adaptive frame/field decoding* processing. The division of a *slice* into macroblock pairs is a *partitioning*.

3.62 **macroblock pair address**: The *raster scan order* number of a *macroblock pair* starting with zero for the top left corner in a *frame*.

3.63 **motion vector**: A two-dimensional vector used for *inter prediction* that provides an offset from the coordinate positions in the *decoded picture* to the coordinates in a *reference picture*.

3.64 **NAL unit**: A syntax structure containing an indication of the type of data to follow and bytes containing that data in the form of an RBSP interspersed as necessary with *emulation prevention bytes*.

3.65 **NAL unit stream**: A sequence of *NAL units* containing the syntax structures associated with the coded video content.

3.66 **network abstraction layer (NAL)**: A definition of syntax structures and additional information including framing and timing that are supported by a system.

3.67 **non-reference picture**: a *decoded picture* that is marked as not used for *inter prediction*. [Ed.Note: this should be changed – How?]

3.68 **opposite parity**: The *opposite parity* of *top* is *bottom*, and vice versa.

3.69 **output order**: The order in which the *decoded pictures* are intended for output.

3.70 **output reordering delay**: A delay between decoding a *coded picture* and output of the *decoded picture* that is caused when the order of pictures specified for output is different from the order specified for decoding.

3.71 **P slice**: A *predictive slice*.

3.72 **parity**: The *parity* of a *field* can be *top* or *bottom*.

3.73 **partitioning**: The division of a set into subsets such that each element of the set is in exactly one of the subsets.

3.74 **picture**: A collective term for a *field* or a *frame*.

3.75 **picture order count**: *Picture* position in output order, relative to the latest previous IDR *picture* in *decoding order*.

3.76 **picture reordering**: The process of reordering the decoded *pictures* when the *decoding order* is different from the *output order*.

3.77 **prediction**: An embodiment of the *prediction process*.

3.78 **prediction process**: The use of a *predictor* to provide an estimate of the sampl e value or data element currently being decoded.

3.79 **prediction residual**: The difference between the represented value of a sample or data element and its *predictor*.

3.80 **predictive slice:** A *slice* that is coded using *inter prediction* from previously-decoded *reference pictures*, using at most one *motion vector* and *reference picture index* to *predict* the sample values of each *block*.

3.81 **predictor**: A combination of previously decoded sample values or data elements used in the *decoding process* of subsequent sample values or data elements.

3.82 **primary coded picture**: a primary coded representation of a *picture*.

3.83 **primary coded slice**: a *slice* belonging to a *primary coded picture*.

3.84 **primary slice data partition**: a *slice data partition* belonging to a *primary coded picture*.

3.85 **probability model**: The set of probability distributions used by the arithmetic decoding process when decoding a symbol. The *context* determines which probability distribution is to be used when decoding a particular symbol at a particular point, block, macroblock, etc. i n the picture. For each symbol, the number of probability distributions in the set is equal to the number of possible values for the *context variable*, i.e., the number of *contexts*.

3.86 **profile**: A specified subset of the syntax of this Recommendation | International Standard.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

3.87 **progressive-scan video:** A form of video content in which each *frame* represents two *fields* which have the same capture time.

3.88 **quantisation parameter**: A parameter used by the *decoding process* for *scaling* of *transform coefficient levels*. [Ed. Note: only mentioned once. Do we need this definition ?]

3.89 **quarter common intermediate format (QCIF)**: A video frame that is 11 *macroblocks* wide and 9 *macroblocks* high.

3.90 **random access**: The ability to start the decoding of a coded *NAL unit stream* at a point other than the beginning of the stream and recover an exact or approximate representation of the *decoded pictures* represented by that *NAL unit stream*.

3.91 **random access point**: A point in the *NAL unit stream* at which random access is possible.

3.92 **raster scan**: A mapping of a rectangular two-dimensional pattern to a one-dimensional pattern such that the first entries in the one-dimensional pattern are from the first top row of the two-dimensional pattern scanned from left to right, followed similarly by the second, third, etc. rows of the pattern (going down) each scanned from left to right.

3.93 **raw byte sequence payload (RBSP):** A syntactical data structure containing an integer number of bytes encapsulated in a *NAL unit* that is either null or has the form of a *string of data bits* containing syntax elements, followed by an *RBSP stop bit*, and followed by zero or more subsequent bits equal to 0.

3.94 **raw byte sequence payload stop bit:** A bit equal to 1 present with in a *raw byte sequence payload (RBSP)* after a *string of data bits*. The location of the end of the *string of data bits* within an *RBSP* can be identified by searching from the end of the *RBSP* for the *RBSP stop bit*, which is the last non-zero bit in the *RBSP*.

3.95 **redundant coded picture**: a redundant coded representation of a *picture* that shall not be used for decoding unless the *primary coded picture* is missing or corrupted. The *decoded redundant picture* [Ed. Note: No such term defined.] may not cover the entire picture area. [Ed. Note: If that's true, then why do we call it a picture – why not just refer to redundant slices?] There should be no noticeable difference between the co-located areas of the *primary decoded picture* [Ed. Note: No such term defined.] and any *redundant decoded slices*. [Ed. Note: No such term defined.]

3.96 **redundant coded slice**: a slice belong to a *redundant coded picture*.

3.97 **redundant slice data partition**: a *slice data partition* belonging to a *redundant coded picture*.

3.98 **reference field**: A *reference field* is used for *inter prediction* when *P*, *SP*, and *B slices* of a *coded field* or a *coded frame* are decoded. See also *reference picture*.

3.99 **reference frame**: A *reference frame* is used for *inter prediction* when *P*, *SP*, and *B slices* of a *coded frame* are decoded. See also *reference picture*.

3.100 **reference index list**: A list of indices that is assigned to the *reference pictures* in the *decoded picture buffer*.

3.101 **reference index list 0**: The list of *reference indices* for use in *reference list 0 prediction* for a *P*, *B*, or *SP slice*. All *inter prediction* used for *P* and *SP* slices is considered *reference list 0 prediction*. The *reference index list 0* is one of *two reference index lists* used for a *B slice*, with the other being the *reference index list 1*.

3.102 **reference list 0 motion vector**: A *motion vector* associated with a *reference index* pointing into the *reference index list 0*.

3.103 **reference list 0 prediction**: *Inter prediction* of the content of a *slice* using a *reference index* into the *reference index list 0*.

3.104 **reference index list 1**: A list of *reference indices* defined for use in *inter prediction* for a *B slice*. The *reference index list1* is one of two lists of *reference indices* used by a *B slice*, with the other being the *reference index list 0*.

3.105 **reference list 1 motion vector**: A *motion vector* associated with a *reference index* pointing into the *reference index list 1*. [Ed. Note: Do we need the word "reference" in this term? How about just "list 1 prediction" ?]

3.106 **reference list 1 prediction**: *Inter prediction* of the content of a *B slice* using a *reference index* into the *reference index list 1*.

3.107 **reference picture**: A *picture* coded with nal_reference_idc not equal to 0. A *reference picture* contains samples that may be used for *inter prediction* of subsequent *pictures* in *decoding order*.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

3.108 **reserved**: The term "reserved", when used in the clauses defining some values of a par ticular syntax element means that these values may be used in extensions of this Recommendation | International Standard by ITU-T | ISO/IEC, and that these values shall not be used unless so specified.

3.109 **residual**: The decoded difference between a *prediction* of a sample or data element and its decoded value.

3.110 **run**: A number of consecutive data elements represented in the decoding process. In one context, the number of zero-valued *transform coefficients* preceding a non-zero *transform coefficient*, in the *block scan* order. In another context, the number of *skipped macroblocks*. [Ed. Note: Do we ever use the first of those two meanings anymore?]

3.111 **sample aspect ratio**: Specifies, for assisting the display process not specified in this Recommendation | International Standard, the ratio between the intended horizontal distance between the rows and the intended vertical distance between the columns of the *luma* sample array in a *frame*. Expressed as *h*:*v*, where *h* is horizontal width and *v* is vertical height (in arbitrary units of spatial distance).

3.112 **scaling**: The process of *scaling* the *transform coefficient levels* resulting in *transform coefficients*.

3.113 **SI slice**: A *switching I slice*.

3.114 **skipped macroblock**: A *macroblock* for which no data is coded other than an indication that the *macroblock* is to be decoded as "skipped". This indication may be common to several *macroblocks*.

3.115 **slice**: An integer number of *macroblocks* or *macroblock pairs* ordered contiguously in *raster scan order* within a particular *slice group*. The division of a *slice group* into slices is a *partitioning*. Although a slice contains *macroblocks* or *macroblock pairs* that are contiguous in raster scan order within a slice group, these *macroblocks* or *macroblock pairs* are not necessarily contiguous within the picture. The addresses of the *macroblocks* are derived from the address of the first *macroblock* or *macroblock pair* in a slice (as represented in the *slice header*) and the *macroblock to slice group map*.

3.116 **slice data partitioning**: A method of partitioning selected syntax elements into syntactical structures based on a *category* associated with each of the syntax elements.

3.117 **slice group**: A subset of the *macroblocks* of a *picture*. The division of the *picture* into slice groups is a *partitioning* of the *picture*. The partition is specified by the *macroblock to slice group map*.

3.118 **slice header**: A part of a *coded slice* containing the *coded representation* of data elements pertaining to the *macroblocks* or *macroblock pairs* represented in the slice.

3.119 **source**: Term used to describe the video material or some of its attributes before encoding.

3.120 **SP slice**: A *switching P slice*.

3.121 **string of data bits (SODB)**: A sequence of some number of bits representing syntax elements present within a *raw byte sequence payload* prior to the *raw byte sequence payload stop bit*.

3.122 **switching I slice**: A *slice* that is coded using prediction only from decoded samples within the same *slice*, encoded such that it can be reconstructed identically to an *SP slice*.

3.123 **switching P slice**: A *slice* that is coded using *inter prediction* from previously-decoded *reference pictures*, using at most one *motion vector* and *reference picture index* to *predict* the sample values of each *block*, encoded such that it can be reconstructed identically to another *SP slice* or an *SI slice*.

3.124 **start code prefix**: A unique sequence of three bytes equal to 0x000001 embedded in the *byte stream format* as a prefix to each *NAL unit*. The location of a *start code prefix* can be used by a decoder to identify the beginning of a new *NAL unit* and the end of a previous *NAL unit*. Emulation of *start code prefixes* is prohibited within *NAL units*.

3.125 **string of data bits (SODB)**: An ordered sequence of some finite number of bits, in which the left-most bit is considered to be the first and most significant bit (MSB) and the right-most bit is considered to be the last and least significant bit (LSB).

3.126 **sub-macroblock**: One quarter of the samples of the samples of a *macroblock*, an 8x8 region aligned to *macroblock* boundaries on both horizontal and vertical edges. [Ed. Note: The second half of that definition needs work – only two of the four edges of a sub-macroblock are aligned to the edges of a macroblock.]

3.127 **symbol**: A *syntax element*, or part thereof, to be decoded.

3.128 **top field**: One of two *fields* that comprise a *frame*. Each row of a *top field* is spatially located immediately above the corresponding row of the *bottom field*.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

3.129 **top macroblock (of a macroblock pair)**: The *macroblock* within a *macroblock pair* that contains the samples in the top row of samples for the *macroblock pair*. For a *field macroblock pair*, the top macroblock represents the samples from the region of the *top field* of the *frame* that lie within the spatial region of the *macroblock pair*. For a *frame macroblock pair*, the top macroblock represents the samples of the *frame* that lie within the bottom half of the spatial region of the *macroblock pair*.

3.130 **transform coefficient**: A scalar quantity, considered to be in a frequency domain, that is associated with a particular two-dimensional frequency index in the *inverse transform* of the decoding process.

3.131 **variable length coding (VLC)**: A reversible procedure for entropy coding that assigns shorter codewords to *symbols* expected to be more frequent and longer codewords to *symbols* expected to be less frequent.

3.132 **XYZ profile decoder**: A decoder able to decode coded data conformin g to the specifications of the XYZ profile (with XYZ being any of the defined Profile names).

3.133 **zig-zag scan**: A specific sequential ordering of *transform coefficients* from (approximately) the lowest spatial frequency to the highest. Zig-zag scan is used for *transform coefficients* in *field macroblocks*

4 Abbreviations

4.1 **ASO**: Arbitrary Slice Order

4.2 **CABAC**: Context-based Adaptive Binary Arithmetic Coding

4.3 **CAT-LB**: Constrained Arrival Time Leaky Bucket

4.4 **CAVLC**: Context-based Adaptive Variable Length Coding

4.5 **CBR**: Constant Bit Rate

4.6 **CIF**: Common Intermediate Format

4.7 **CPB**: Coded Pictures Buffer

4.8 **DPB**: Decoded Picture Buffer

4.9 **FIFO**: First-In, First-Out

4.10 **FMO**: Flexible Macroblock Ordering

4.11 **HRD**: Hypothetical Reference Decoder

4.12 **IDR**: Instantaneous Decoder Refresh

4.13 **LPS**: Least Probable Symbol

4.14 **LSB**: Least Significant Bit

4.15 **MB**: Macroblock

4.16 **MPS**: Most Probable Symbol

4.17 **MSB**: Most Significant Bit

4.18 **NAL**: Network Abstraction Layer

4.19 **QCIF**: Quarter Common Intermediate Format

4.20 **RBSP**: Raw Byte Sequence Payload

4.21 **SEI**: Supplemental Enhancement Information

4.22 **SODB**: String Of Data Bits

4.23 **VBR**: Variable Bit Rate

4.24 **VCL**: Video Coding Layer

4.25 **VLC**: Variable Length Coding

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

5 Conventions

The mathematical operators used in this Specification are similar to those used in the C programming language. However, integer division and arithmetic shift oper ations are specifically defined. Numbering and counting conventions generally begin from 0.

5.1 Arithmetic operators

The following arithmetic operators are defined as follows

+	Addition
–	Subtraction (as a two-argument operator) or negation (as a unary prefi x operator)
×	Multiplication
*	Multiplication
	Multiplication
$a^{\wedge}b$	Exponentiation. Indicates a to the power of b.
a^b	Exponentiation. Indicates a to the power of b. In other contexts, such notation is used for superscripting not intended for interpret ation as exponentiation.
/	Integer division with truncation of the result toward zero. For example, 7/4 and −7/−4 are truncated to 1 and −7/4 and 7/−4 are truncated to −1.
÷	Used to denote division in mathematical equations where no truncation or rounding is intended.
$\frac{a}{b}$	Used to denote division in mathematical equations where no truncation or rounding is intended.
$\sum_{i=a}^{b} f(i)$	The summation of the $f(i)$ with i taking all integer values from a up to and including b.
a % b	Modulus. Remainder of a divided by b, defined only for a and b both positive integers

5.2 Logical operators

The following logical operators are defined as follows

a && b	Boolean logical "and" of a and b
a || b	Boolean logical "or" of a and b
!	Logical NOT

5.3 Relational operators

The following relational operators are defined as follows

>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
==	Equal to
!=	Not equal to

5.4 Bit-wise operators

The following bit-wise operators are defined as follows

&	AND
|	OR
a >> b	Arithmetic right shift of a two's complement i nteger representation of a by b binary digits. This function is defined only for positive values of b. Bits shifted into the MSBs as a result of the right shift shall have a value equal to the MSB of a prior to the shift operation.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

$a << b$ Arithmetic left shift of a two's complement integer representation of a by b binary digits. This function is defined only for positive values of b. Bits shifted into the LSBs as a result of the left shift have a value equal to 0.

5.5 Assignment

The following arithmetic operators are defined as follows

= Assignment operator

$x = a..b$ x takes on values starting from a to b inclusive, with x, a, and b being integer numbers. [Ed. Note: Is this really an assignment operator (I don't think so), and is it really used anywhere?]

++ Increment, i.e. x++ is equivalent to $x = x + 1$; when used in an array index, evaluates to the value of the variable prior to the increment operation.

-- Decrement, i.e. x-- is equivalent to $x = x - 1$; when used in an array index, evaluates to the valu e of the variable prior to the decrement operation.

+= Increment by amount specified, i.e., x += 3 is equivalent to x = x + 3, and x += (-3) is equivalent to x = x + (-3).

-= Decrement by amount specified, i.e., x -= 3 is equivalent to x = x - 3, and x -= (-3) is equivalent to x = x - (-3).

5.6 Mathematical functions

The following mathematical functions are defined as follows

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases} \tag{5-1}$$

Ceil(x) rounds x up to the nearest integer greater than or equal to x. (5-2)

$$\text{Clip1}(x) = \text{Clip3}(0, 255, x) \tag{5-3}$$

$$\text{Clip3}(a, b, c) = \begin{cases} a & ; \quad c < a \\ b & ; \quad c > b \\ c & ; \quad \text{otherwise} \end{cases} \tag{5-4}$$

$$\text{InverseRasterScan}(a, b, c, d, e) = \begin{cases} (a\%(d/b))*b; & e == 0 \\ (a/(d/b))*c; & e == 1 \end{cases} \tag{5-5}$$

Log2(x) returns the base-2 logarithm of x. (5-6)

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases} \tag{5-7}$$

$$\text{RasterScan}(x, y, N_x) = x + y * N_x$$

$$\text{RasterTo4x4LumaBlkOffset}(x) = \{ [(x \% 4) / 2] - [(x / 4) \% 2] \} * 2 \tag{5-8}$$

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x \geq 0 \\ -1 & ; \quad x < 0 \end{cases} \tag{5-9}$$

$$\text{4x4LumaBlkScan}(x, y) = (x / 2) * 4 + (y / 2) * 8 + \text{RasterScan}(x \,\%\, 2, y \,\%\, 2, 2) \tag{5-10}$$

[Ed. Note: I don't like starting variable names with numerical characters – and it 's illegal to do that in C.]

6 Source, coded, decoded, and output data formats

6.1 Bitstream formats

This subclause specifies the relationship between the NAL unit stream and byte stream, eith er of which are referred to as the bitstream.

The bitstream can be in one of two formats: the NAL unit stream format or the byte stream format. The NAL unit stream format is conceptually the more "basic" type. It consists of a sequence of syntax structure s called NAL units. This sequence is ordered in decoding order. There are constraints imposed on the decoding order (and contents) of the NAL units in the NAL unit stream.

The byte stream format can be constructed from the NAL unit stream format by orde ring the NAL units in decoding order and prefixing each NAL unit with a start code prefix and zero or more zero -valued bytes to form a stream of bytes. The NAL unit stream format can be extracted from the byte stream format by searching for the location o f the unique start code prefix pattern within this stream of bytes. Methods of framing the NAL units in a manner other than use of the byte stream format are outside the scope of this Recommendation | International Standard. The byte stream format is specified in Annex B.

6.2 Source, decoded, and output picture formats

This subclause specifies the relationship between source and decoded frames and fields that is given via the bitstream.

The video source that is represented by the bitstream is a sequence of frames and/or fields (called collectively pictures) in decoding order.

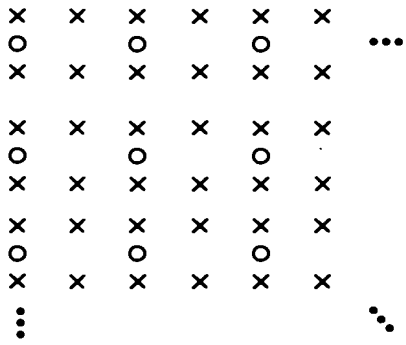
The source and decoded pictures (frames or fields) are each comprised of three sample arrays, one luma and two chroma sample arrays.

The width and height of the luma arrays are multiples of 16 samples. This Recommendation | International Standard represents colour sequences using 4:2:0 chroma sampling. I.e. the width and height of each of the two chroma samples ' arrays are half of the width and height of the lum a array of the same picture. The width and height of chroma arrays are multiple of 8 samples. The height of a luma array that is coded as two separate fields or in macroblock -based frame field adaptive mode (see below) is a multiple of 32 samples. The heig ht of each chroma array that is coded as two separate fields or in macroblock-based frame field adaptive mode (see below) is a multiple of 16 samples. The width or height of pictures output from the decoder need not be a multiple of 16 and can be specified using a cropping rectangle.

The width of fields coded referring to a specific sequence parameter set is the same as that of frames coded referring to the same sequence parameter set (see below). The height of fields coded referring to a specific sequence parameter set, is half that of frames coded referring to the same sequence parameter set (see below).

The nominal vertical and horizontal relative locations of luma and chroma samples in frames are shown in Figure 6-1. Alternative chroma sample relative locations may be indicated in video usability information (see Annex E).

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)



Guide:

× = Location of luminance sample

○ = Location of chrominance sample

Figure 6-1 – Nominal vertical and horizontal locations of 4:2:0 luma and chroma samples in a frame

This Recommendation | International Standard specifies a syntax and decoding process for video that originated in either progressive-scan or interlaced-scan form, which may be mixed together in the same sequence.

A video frame consists of two fields as described below. A coded picture may represent a frame or an individual field. A video sequence conforming to this Recommendation | International Standard may contain arbitrary combinations of coded frames and coded fields. The decoding process is also specified in a manner that allows smaller regions of a coded frame to be coded as a representation either as frame or field region.

Source and decoded fields are one of two types: top field or bottom field. The nominal vertical and horizontal relative locations of luma and chroma samples in fields are shown in Figure 6-2. Alternative chroma sample relative locations may be indicated in the video usability information (see Annex E).

When two fields are output at the same time, or combined to be used as a reference frame (see below), the two fields (which must be of opposite parity) are interleaved. The first (i.e., top), third, fifth, etc. rows of a decoded frame are the top field rows. The second, fourth, sixth, etc. rows of a decoded frame are the bottom field rows. A top field picture consists of only the top field rows of a decoded frame. When the top field or bottom field of a frame is used as a reference field (see below) only the even rows (for a top field) or the odd rows (for a bottom field) of the frame are used.

The two decoded fields of an interlaced frame are separated in time. They may be decoded separately as two fields or together as a frame.

NOTE - A progressive frame should always be coded as a single frame picture. However, a progressive frame is still considered to consist of two fields (at the same instant in time) and may be represented using coded fields.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

The nominal vertical and horizontal relative locations of luma and chroma samples in interlaced fields are shown in Figure 6-2. The vertical sampling relative locations of the chroma samples in a top field of an interlaced frame are specified as shifted up by 1/4 luma sample height relative to the field -sampling grid in order for these samples to align vertically to the usual location relative to the full -frame sampling grid. The vertical sampling locations of the chroma samples in a bottom field of an interlaced frame are specified as shifted down by 1/4 luma sample height relative to the field-sampling grid in order for these samples to align vertically to the usual location relative to the full -frame sampling grid. The horizontal sampling relative locations of the chroma samples are specified as unaffected by the application of interlaced field coding.

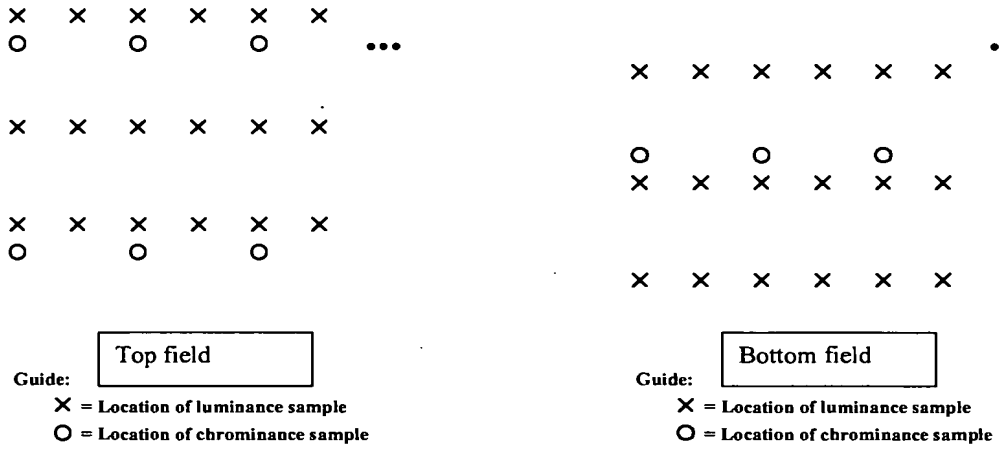


Figure 6-2 – Nominal vertical and temporal sampling locations of samples in 4:2:0 interlaced frames

6.3 Spatial subdivision of pictures, slices, macroblocks, and sub-macroblocks

This subclause specifies how a picture is partitioned into slices and macroblocks. Pictures are divided into slices. A slice is a sequence of macroblocks or macroblock pairs. [Ed. Note: Need to integrate Horowitz input for MB -AFF]

Slices are partitioned into macroblocks. Each macroblock of a picture is in no more than one slice. [Ed. Note: Work on redundant pictures – in a redundant picture, some macroblocks may not be represented in any slice, but in primary coded pictures each macroblock of the pi cture shall be in some slice.]

Each macroblock is comprised of a one 16x16 luma and two 8x8 chroma sample arrays. When macroblock -adaptive frame/field decoding is not in use, each macroblock represents a spatial rectangular region of the picture. For ins tance, a QCIF picture may be divided into two slices as shown in Figure 6-3. If the first slice contains 48 macroblocks, then the second slice contains 51 macroblocks, since the entire QCIF picture is divided into 99 macroblocks as shown in Figure 6-3.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

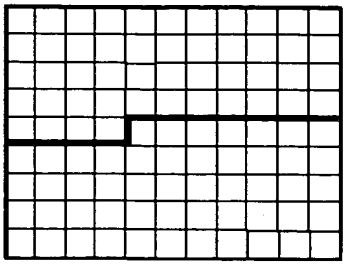


Figure 6-3 – A picture with 11 by 9 macroblocks (QCIF picture)

A macroblock location in the picture is given by a pair of indices where the first index is the macroblock column index starting with 0 for the leftmost column, and the second index is the macroblock row index starting with 0 for the topmost row. [Ed. Note: Not true for MB-AFF. Much of this description is redundant with the definitions section.]

When macroblock-adaptive frame/field decoding is in use, the picture is partitioned into slices containing an integer number of macroblock pairs as shown in Figure 6-4. Each macroblock pair consists of two macroblocks.

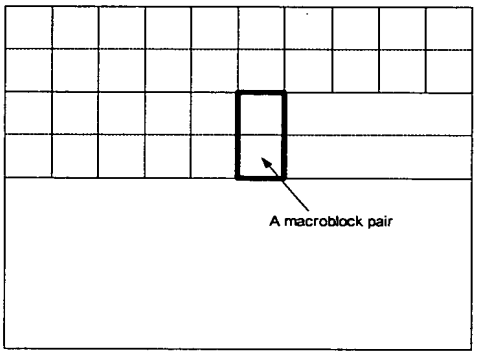


Figure 6-4 – Partitioning of the decoded frame into macroblock pairs.

Macroblocks or sub-macroblocks may be partitioned and the paritions are scanned for inter prediction as shown in Figure 6-5. The outer quadrats refer to the samples in a macroblock or sub-macroblock, respectively. The rectangles or quadrats refer to the partitions. The numbers in eacb quadrat or rectangle specify the index of the inverse macroblock partition scan or inverse sub-macroblock partition scan.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

1 m
16'
asso

Figure 6-5 – Macroblock partitions, sub-macroblock partitions, inverse macroblock partition scans, and inverse sub-macroblock partition scans. [Ed. Note: Figure does not print properly on at least one printer.]

6.4 Inverse scanning processes

This subclause specifes inverse scanning processes, i.e., the mapping of indices to locations.

6.4.1 Inverse macroblock scanning process

[Ed.Note: the use of macroblock addresses may make the use of macroblock indices not necessary]

Input to this process is the macroblock index MbIdx.

Output of this process is the location of the upper -left luma sample (MbLumaLoc) for the macroblock with index MbIdx relative to the upper-left sample of the picture.

The variable SliceGroupMbIdxList[MbIdx] specifies the macroblock index in a scan that assumes one slice group and one slice in the picture. [Ed.Note: specify how this variable is obtained].

The inverse macroblock scann ing process is specified by

PartIdx = SliceGroupMbIdxList[MbIdx]

Width = (frame_width_in_mbs_minus1 + 1) * 16

MbLumaLoc[MbIdx][0] = InverseRasterScan(PartIdx, 16, 16, Width, 0)

MbLumaLoc[MbIdx][1] = InverseRasterScan(PartIdx, 16, 16, Width, 1)

6.4.2 Inverse macroblock pair scanning process

[Ed.Note: the use of macroblock pair addresses may make the use of macroblock pair indices not necessary]

Input to this process is the macroblock index MbPairIdx.

Output of this process is the location of the u pper-left luma sample (MbPairLumaLoc) for the macroblock pair with index MbPairIdx relative to the upper -left sample of the picture.

The variable SliceGroupMbPairIdxList[MbPairIdx] specifies the macroblock pair index in a scan that assumes one slice group and one slice in the picture. [Ed.Note: specify how this variable is obtained].

The inverse macroblock scanning process is specified by

PartIdx = SliceGroupMbPairIdxList[MbPairIdx]

Width = (frame_width_in_mbs_minus1 + 1) * 16

MbPairLumaLoc[MbPairIdx][0] = InverseRasterScan(PartIdx, 16, 16, Width, 0)

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

MbPairLumaLoc[MbPairIdx][1] = InverseRasterScan(PartIdx, 16, 16, Width, 1)

6.4.3 Inverse macroblock in macroblock pair scanning process

[Ed.Note: the use of macroblock addresses may make the use of macroblock indices not necessary]

Input to this process is the macroblock index MbInMbPairIdx.

Output of this process is the location of the upper-left luma sample (MbLumaLoc) for the macroblock pair with index MbInMbPairIdx relative to the upper-left sample of the macroblock pair.

The inverse macroblock in macroblock pair scanning process is specified by

MbInMbPairLumaLoc[MbInMbPairIdx][0] = InverseRasterScan(MbInMbPairIdx, 16, 16, 16, 0)

MbInMbPairLumaLoc[MbInMbPairIdx][1] = InverseRasterScan(MbInMbPairIdx, 16, 16, 16, 1)

NOTE - The inverse macroblock in macroblock pair scanning process is independent of whether the macroblock pair is coded in frame or field decoding mode.

The macroblock with index 0 is referred to as the top-macroblock, the macroblock with index 1 is referred to as the bottom-macroblock.

6.4.4 Inverse macroblock partition scanning process [Ed.Note: may not be needed as separate scan spec]

Input to this process is the index of the macroblock partition MbPartIdx, the width of the mac roblock partition MbPartWidth, and the height of the macroblock partition MbPartHeight.

Output of this process is the location of the upper-left luma sample (MbPartLumaLoc) for the macroblock partition with index MbPartIdx relative to the upper-left sample of the macroblock.

The inverse macroblock partition scanning process is specified by

MbPartLumaLoc[MbPartIdx][0] = InverseRasterScan(MbPartIdx, MbPartWidth, MbPartHeight, 16, 0)

MbPartLumaLoc[MbPartIdx][1] = InverseRasterScan(MbPartIdx, MbPart Width, MbPartHeight, 16, 1)

[Ed.Note: we may need to specify MbPartWidth and MbPartHeight for 16x8 and 8x16 to be clear]

6.4.5 Inverse sub-macroblock partition scanning process [Ed.Note: may not be needed as separate scan spec]

Input to this process is the index of the sub-macroblock partition SubMbPartIdx, the width of the sub-macroblock partition MbPartWidth, and the height of the sub-macroblock partition SubMbPartHeight.

Output of this process is the location of the upper-left luma sample (SubMbPartLumaLoc) for the sub-macroblock partition with index SubMbPartIdx relative to the upper-left sample of the sub-macroblock.

The inverse sub-macroblock partition scanning process is specified by

SubMbPartLumaLoc[MbPartIdx][0] = InverseRasterScan(SubMbPartIdx, SubMbPartWidth, SubMbPartHeight, 8, 0)

SubMbPartLumaLoc[MbPartIdx][1] = InverseRasterScan(SubMbPartIdx, SubMbPartWidth, SubMbPartHeight, 8, 1)

[Ed.Note: we may need to specify SubMbPartWidth and SubMbPartHeight for 8x4 and 4x8 to be clear]

6.4.6 Inverse 4x4 luma block scanning process

Input to this process is the index of the 4x4 luma block 4x4LumaBlkIdx.

Output of this process is the location of the upper-left luma sample (4x4LumaBlkLoc) for the 4x4 luma block with index 4x4LumaBlkIdx relative to the upper-left sample of the macroblock.

The inverse 4x4 luma block scanning process is specified by

4x4LumaBlkLoc[4x4LumaBlkIdx][0] = InverseRasterScan(4x4LumaBlkIdx / 4, 8, 8, 16, 0) +
InverseRasterScan(4x4LumaBlkIdx % 4, 4, 4, 8, 0)

4x4LumaBlkLoc[4x4LumaBlkIdx][1] = InverseRasterScan(4x4LumaBlkIdx / 4, 8, 8, 16, 1) +
InverseRasterScan(4x4LumaBlkIdx % 4, 4, 4, 8, 1)

6.4.7 Derivation process of the availability for neighbouring macroblock addresses

Inputs to this process are a macroblock address (MbAddress).

Output of this process is the availability of the macroblock MbAddress.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

The macroblock is marked available, unless one of the following conditions is true in which case the macroblock shall be marked as not available:

MbAddress < 0

MbAddress > MaxMbAddress – 1

the macroblock with address MbAddress belongs to a different slice

the macroblock with address MbAddress is at a later position in the decoding order.

6.4.8 Derivation process of the availability for neighbouring macroblock pair addr esses

Inputs to this process are a macroblock pair address (MbPairAddress).

Output of this process is the availability of the macroblock pair MbPairAddress.

The macroblock pair is marked available, unless one of the following conditions is true in which ca se the macroblock pair shall be marked as being not available:

MbPairAddress < 0

MbPairAddress > MaxMbPairAddress – 1

the macroblock pair with address MbPairAddress belongs to a different slice

the macroblock pair with address MbPairAddress is at a lat er position in the decoding order.

7 Syntax and semantics

7.1 Method of describing the syntax in tabular form

The syntax is described in a manner that closely follows the C -language syntactic constructs. Syntax elements in the bitstream are represented in **bold** type. Each syntax element is described by its name (all lower case letters with underscore character to separae, as needed, groups of letters), its syntax category and descriptor for its method of coded representation. A decoder behaves according to the val ue of the syntax element and on the values of previously decoded syntax elements. When a value of a syntax element is used in the syntax tables or the text, it apears in regular (i.e. not bold type). In some cases the syntax tables may use the values of ot her varaibales derived from syntax elements values. Such variables appear in the syntax tables, or text, named by a mixture of lower case and upper case letters, starting with an upper case letter, and without any underscore characters. The derivation is s pecified in the text. In some cases, "mnemonic" names for syntax elements or variable are used interchangeably with their numerical values. The association of values and names is specified in the text. The names are constructed from one or more groups of l etters separated by an underscore character. Each group starts with an upper case letter and may contain more upper case letters.

Hexadecimal notation may be used instead of binary notation when the number of bit is a multiple of 4.

Square parentheses are used for indexing.

The syntax tables describe a superset of the syntax of all correct and error -free input bitstreams. [Ed.Note: remove error or change to NOTE] Additional constraints on the syntax form may be specified in other clauses. An actual decoder must implement correct means for identifying entry points into the bitstream for proper decoding and to identify and handle errors in the bitstream [Ed.Note: remove error or change to NOTE]. The methods for identifying and handling errors and other such s ituations are not described here.

Following C-language conventions, a value equal to 0 represents a FALSE condition in a test statement. The value TRUE is represented by 1, but any other value different than zero is also understood as TRUE.

The following table lists examples of pseudo code used to describe the syntax. When syntax_element appears, it indicates that a data element is read (extracted) from the bitstream and the bitstream pointer advances to the bit following the last bit of the data element ex tracted.

[Ed.Note: specify function of functions, specify passing of parameters, explain bold vs. not bold syntax elements, specify variables PayloadType ..., mention meaning of vertical bar in category and descriptor]

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

	Category	Descriptor
/* A statement can be a syntax element with an associated syntax category and descriptor or can be an expression used to specify conditions for the existence, type, and quantity of syntax elements, as in the following two examples */		
syntax_element	3	ue(v)
conditioning statement		
/* A group of statements enclosed in curly brackets is a compound statement and is treated functionally as a single statement. */		
{		
statement		
statement		
...		
}		
/* A "while" structure indicates a test of whether a cond ition is true, and if true, indicates evaluation of a statement (or compound statement) repeatedly until the condition is no longer true */		
while(condition)		
statement		
/* A "do ... while" structure indicates evaluation of a statement once, followed by a test of whether a condition is true, and if true, indicates repeated evaluation of the statement until the condition is no longer true */		
do		
statement		
while(condition)		
/* An "if ... else" structure indicates a test of whether a condition is true, and if the condition is true, indicates evaluation of a primary statement, otherwise indicates evaluation of an alternative statement. The "else" part of the structure and the associated alternative statement is omitted if no alternative statement evaluation is needed */		
if(condition)		
primary statement		
else		
alternative statement		
/* A "for" structure indicates evaluation of an initial statement, followed by a test of a condition, and if the condition is true, indica tes repeated evaluation of a primary statement followed by a subsequent statement until the condition is no longer true. */		
for(initial statement; condition; subsequent statement)		
primary statement		

7.2 Definitions of functions, categories and descriptors

The functions presented here are used in the syntactical description. These functions assume the existence of a bitstream pointer with an indication of the position of the next bit to be read by the decoder from the bitstream. Function names are constructed as syntax element names with left and right round parentheses including zero or more parameter names (for definition) or values (for usage), separated by commas (if more than one parameter). [Ed.Note: fix bitstream vs. byte stream vs. coded stream vs. NAL unit stream]

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

byte_aligned( )

- Returns TRUE if the current position in the bitstream is on a byte boundary, i.e., the next bit in the bitstream is the first bit in a byte. Otherwise it returns FALSE.

more_data_in_byte_stream( )

- Returns TRUE if more data follows in the byte stream. Otherwise it returns FALSE. Used only in the byte stream NAL unit syntax structure specified in Annex B.

more_rbsp_data( )

- Returns TRUE if there is more data in an RBSP before rbsp_trailing_bits( ). Otherwise it returns FALSE. The method for enabling determination of whether there is more data in the RBSP is specified by the system (or in Annex B for systems that use the byte stream format).

more_rbsp_trailing_data( )

- Returns TRUE if there is more data in an RBSP. Other wise it returns FALSE.

next_bits(*n*)

- Provides the next bits in the bitstream as an unsigned integer value for comparison purposes, without advancing the bitstream pointer. Provides a look at the next *n* bits in the bitstream with *n* being its argument. If used within the byte stream NAL unit syntax structure specified in Annex B, returns a value of 0 if fewer than *n* bits remain within the byte stream.

next_mb_address(*i*)

- Provides the macroblock address of the next macroblock in the slice group after the ma croblock with macroblock address *i* in macroblock address order.

read_bits(*n*)

- Reads the next *n* bits from the bitstream and advances the bitstream pointer by *n* bit positions. If *n* is equal to 0, then read_bits(*n*) is defined to return a value equal to 0 and to not advance the bitstream pointer.

total_coeff( )

- Returns the number of coefficients from coeff_token. See subclause 7.3.5.3.1.

trailing_ones( )

- Returns the trailing ones from coeff_token. See subclause 7.3.5.3.1.

Categories (labelled in the table as C) specify the slice data partition for slice data into three partitions. Partition A contains all syntax elements of category 4. Partition B contains all syntax el ements of category 5. Partition C contains all syntax elements of Category 6. The meaning of other category values is not specified.

The following descriptors specify the parsing process of each syntax element. For some syntax elements, two descriptors, separated by a vertical bar, are used. In these cases, the left descriptors apply when entropy_coding_mode is equal to 0 and the right descriptor applies when entropy_coding_mode is equal to 1.

- **ae(v)**: context-adaptive arithmetic entropy-coded syntax element. The parsing process for this descriptor is specified in subclause 9.3.
- **b(8)**: byte having any pattern of bit string (8 bits). The parsing process for this descriptor is specified by the return value of the function read_bits (8).
- **ce(v)**: context-adaptive variable-length entropy-coded syntax element with the left bit first. The parsing process for this descriptor is specified in subclause 9.2.
- **f(*n*)**: fixed-pattern bit string using *n* bits written (from left to right) with the left bit first. The parsing process for this descriptor is specified by the return value of the function read_bits(*n*).
- **i(*n*)**: signed integer using *n* bits. If *n* is "v", the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this descriptor is specified by the return value of the function read_bits(*n*) interpreted as a two's complement representation with most significant bit written first.
- **se(v)**: signed integer Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in subclause 9.1.
- **te(v)**: truncated Exp-Golomb-coded syntax element with left bit first. The parsing process for this descriptor is specified in subclause 9.1.

Formatted: Indent: Left: 0.25", Bulleted + Level: 1 + Aligned at: 0.5" + Tab after: 0.75" + Indent at: 0.75", Tabs: 0.25", List tab

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

- **u(*n*)**: unsigned integer using *n* bits. If *n* is "v", the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this descriptor is specified by the return value of the function read_bits(*n*) interpreted as a binary representation of an unsigned integer with most significant bit written first.
- **ue(v)**: unsigned integer Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in subclause 9.1.

7.3 Syntax in tabular form

7.3.1 NAL unit syntax

nal_unit(NumBytesInNALunit) {	C	Descriptor
forbidden_bit		f(1)
nal_reference_idc		u(2)
nal_unit_type		u(5)
NumBytesInRBSP = 0		
for(i = 0; i < NumBytesInNALunit - 1; i++) {		
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {		
rbsp[NumBytesInRBSP++ **]**		b(8)
rbsp[NumBytesInRBSP++ **]**		b(8)
i += 2		
emulation_prevention_byte /* equal to 0x03 */		f(8)
} else		
rbsp[NumBytesInRBSP++ **]**		b(8)
}		
}		

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

7.3.2 Raw byte sequence payloads and RBSP trailing bits syntax

7.3.2.1 Sequence parameter set RBSP syntax

seq_parameter_set_rbsp( ) {	C	Descriptor
profile_idc	0	ue(v)
level_idc	0	ue(v)
seq_parameter_set_id	0	ue(v)
log2_max_frame_num_minus4	0	ue(v)
pic_order_cnt_type	0	ue(v)
if(pic_order_cnt_type == 1)		
num_ref_frames_in_pic_order_cnt_cycle	0	ue(v)
if(pic_order_cnt_type < 2) {		
delta_pic_order_always_zero_flag	0	u(1)
offset_for_non_ref_pic	0	se(v)
offset_for_top_to_bottom_field	0	se(v)
}		
if(pic_order_cnt_type == 0)		
offset_for_ref_frame[0]	0	se(v)
else if(pic_order_cnt_type == 1) {		
for(i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++)		
offset_for_ref_frame[i]	0	se(v)
}		
num_ref_frames	0	ue(v)
required_frame_num_update_behaviour_flag	0	u(1)
frame_width_in_mbs_minus1	0	ue(v)
frame_height_in_mbs_minus1	0	ue(v)
frame_coding_only_flag	0	u(1)
mb_adaptive_frame_field_flag	0	u(1)
direct_temporal_constrained_flag	0	u(1)
vui_seq_parameters_present_flag	0	u(1)
if(vui_seq_parameters_present_flag)		
vui_seq_parameters( )	0	
rbsp_trailing_bits( )	0	
}		

[Ed.Note: Syntax structure functions should have category assignments , mention that parameters of parameter sets are global]

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

7.3.2.2 Picture parameter set RBSP syntax

pic_parameter_set_rbsp( ) {	C	Descriptor
pic_parameter_set_id	1	ue(v)
seq_parameter_set_id	1	ue(v)
entropy_coding_mode	1	ue(v)
if(pic_order_cnt_type < 2)		
pic_order_present_flag	1	u(1)
num_slice_groups_minus1	1	ue(v)
if(num_slice_groups_minus1 > 0) {		
mb_slice_group_map_type	1	ue(v)
if(mb_slice_group_map_type = = 0)		
for(i = 0; i <= num_slice_groups_minus1; i++)		
run_length[i]	1	ue(v)
else if(mb_slice_group_map_type = = 2)		
for(i = 0; i < num_slice_groups_minus1; i++) {		
top_left_mb[i] [Ed.Note: broken* !!!]	1	u(v)
bottom_right_mb[i] [Ed.Note: broken* !!!]	1	u(v)
}		
else if(mb_slice_group_map_type = = 3 || mb_slice_group_map_type = = 4 || mb_slice_group_map_type = = 5) {		
slice_group_change_direction_flag	1	u(1)
slice_group_change_rate_minus1	1	ue(v)
} else if(mb_slice_group_map_type = = 6)		
for(i = 0; i < MaxMbAddress [Ed.Note: still a zombie*, broken !!!]; i++)		
slice_group_id	1	u(v)
}		
num_ref_idx_l0_active_minus1	1	ue(v)
num_ref_idx_l1_active_minus1	1	ue(v)
weighted_pred_flag	1	u(1)
weighted_bipred_explicit_flag	1	u(1)
weighted_bipred_implicit_flag	1	u(1)
pic_init_qp_minus26 /* relative to 26 */	1	se(v)
pic_init_qs_minus26 /* relative to 26 */	1	se(v)
chroma_qp_index_offset	1	se(v)
deblocking_filter_parameters_present_flag	1	u(1)
constrained_intra_pred_flag	1	u(1)
redundant_slice_flag	1	u(1)
vui_pic_parameters_flag	1	u(1)
if(vui_pic_parameters_flag) {		
vui_pic_parameters( )	1	
}		
rbsp_trailing_bits( )	1	
}		

[Ed.Note: *you don't know whether this pps is for frame or field ! This depen ds on MaxMbAddress which itself depends on whether you code frame or field !]

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

7.3.2.3 Supplemental enhancement information RBSP syntax

sei_rbsp( ) {	C	Descriptor
do		
sei_message( )	7	
while(more_rbsp_data( ))		
rbsp_trailing_bits( )	7	
}		

7.3.2.3.1 Supplemental enhancement information message syntax

sei_message( ) {	C	Descriptor
PayloadType = 0		
while(next_bits(8) == 0xFF) {		
byte_ff /* equal to 0xFF */	7	f(8)
PayloadType += 255		
}		
last_payload_type_byte	7	u(8)
PayloadType += last_payload_typ e_byte		
PayloadSize = 0		
while(next_bits(8) == 0xFF) {		
byte_ff /* equal to 0xFF */	7	f(8)
PayloadSize += 255		
}		
last_payload_size_byte	7	u(8)
PayloadSize += last_payload_size_byte		
sei_payload(PayloadType, PayloadSize)	7	
}		

7.3.2.4 Picture delimiter RBSP syntax

pic_delimiter_rbsp( ) {	C	Descriptor
pic_type	8	u(3)
rbsp_trailing_bits( )	8	
}		

7.3.2.5 Filler data RBSP syntax

filler_data_rbsp(NumBytesInRBSP) {	C	Descriptor
while(next_bits(8) == 0xFF)		
byte_ff /* equal to 0xFF */	9	f(8)
rbsp_trailing_bits( )	9	
}		

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

7.3.2.6 Slice layer RBSP syntax

slice_layer_no_partitioning_rbsp( ) {	C	Descriptor
slice_header( )	4	
slice_data( ) /* all categories of slice_data( ) syntax */	4 | 5 | 6	
rbsp_slice_trailing_bits( )	4	
}		

7.3.2.7 Slice data partition RBSP syntax

7.3.2.7.1 Slice data partition A RBSP syntax

slice_data_partition_a_layer_rbsp( ) {	C	Descriptor
slice_header( )	4	
slice_id	4	ue(v)
slice_data( ) /* only the category 4 parts of slice_data( ) syntax */	4	
rbsp_slice_trailing_bits( )	4	
}		

7.3.2.7.2 Slice data partition B RBSP syntax

slice_data_partition_b_layer_rbsp( ) {	C	Descriptor
slice_id	5	ue(v)
if(redundant_slice_flag)		
redundant_pic_cnt	5	ue(v)
slice_data( ) /* only the category 5 parts of slice_data( ) syntax */	5	
rbsp_slice_trailing_bits( )	5	
}		

7.3.2.7.3 Slice data partition C RBSP syntax

slice_data_partition_c_layer_rbsp( ) {	C	Descriptor
slice_id	6	ue(v)
if(redundant_slice_flag)		
redundant_pic_cnt	5	ue(v)
slice_data( ) /* only the category 6 parts of slice_data( ) syntax */	6	
rbsp_slice_trailing_bits( )	6	
}		

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

7.3.2.8 RBSP slice trailing bits syntax

rbsp_slice_trailing_bits( ) {	C	Descriptor
rbsp_trailing_bits( )	All	
if(entropy_coding_mode == 1)		
while(more_rbsp_trailing_data( ))		
cabac_zero_word /* equal to 0x0000 */	All	f(16)
}		

7.3.2.9 RBSP trailing bits syntax

rbsp_trailing_bits( ) {	C	Descriptor
rbsp_stop_bit /* equal to 1 */	All	f(1)
while(!byte_aligned( ))		
rbsp_alignment_bit /* equal to 0 */	All	f(1)
}		

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

7.3.3 Slice header syntax

slice_header( ) {	C	Descriptor
pic_parameter_set_id	4	ue(v)
frame_num	4	u(v)
if(!frame_coding_only_flag) {		
field_pic_flag	4	u(1)
if(field_pic_flag)		
bottom_field_flag	4	u(1)
}		
first_mb_in_slice	4	u(v)
slice_type	4	ue(v)
if(pic_order_cnt_type < 2 && !delta_pic_order_always_zero_flag) {		
delta_pic_order_cnt[0]	4	se(v)
if(pic_order_present_flag == 1 && !field_pic_flag)		
delta_pic_order_cnt[1]	4	se(v)
}		
if(redundant_slice_flag)		
redundant_pic_cnt	4	ue(v)
if(slice_type == B)		
direct_spatial_mv_pred_flag	4	u(1)
num_ref_idx_active_override_flag	4	u(1)
if(num_ref_idx_active_override_flag) {		
if(slice_type == P || slice_type == SP || slice_type == B) {		
num_ref_idx_l0_active_minus1	4	ue(v)
if(slice_type == B)		
num_ref_idx_l1_active_minus1	4	ue(v)
}		
}		
ref_idx_reordering( )	4	
if((weighted_pred_flag && (slice_type == P || slice_type == SP)) || (weighted_bipred_explicit_flag && slice_type == B))		
pred_weight_table( )	4	
if(nal_unit_type == 5)		
non_ref_pic_reset_flag	4	u(1)
else		
dec_pic_buffer_management( )	4	
slice_qp_delta	4	se(v)
if(slice_type == SP || slice_type == SI) {		
if(slice_type == SP)		
sp_for_switch_flag	4	u(1)
slice_qs_delta	4	se(v)
}		
if(deblocking_filter_parameters_present_flag == 1) {		
disable_deblocking_filter_idc	4	ue(v)
if(disable_deblocking_filter_idc != 1) {		
slice_alpha_c0_offset_div2	4	se(v)
slice_beta_offset_div2	4	se(v)
}		

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

}		
if(num_slice_groups_minus1 > 0 && mb_slice_group_map_type >= 3 && mb_slice_group_map_type <= 5)		
slice_group_change_cycle	4	u(v)
}		

7.3.3.1 Reference index reordering syntax

ref_idx_reordering( ) {	C	Descriptor
if(slice_type != I && slice_type != SI) {		
ref_idx_reordering_flag_l0	4	u(1)
if(ref_idx_reordering_flag_l0) {		
do {		
remapping_of_pic_nums_idc	4	ue(v)
if(remapping_of_pic_nums_idc == 0 || remapping_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	4	ue(v)
else if(remapping_of_pic_nums_idc == 2)		
long_term_pic_idx	4	ue(v)
} while(remapping_of_pic_nums_idc != 3)		
}		
}		
if(slice_type == B) {		
ref_idx_reordering_flag_l1	4	u(1)
if(ref_idx_reordering_flag_l1) {		
do {		
remapping_of_pic_nums_idc	4	ue(v)
if(remapping_of_pic_nums_idc == 0 || remapping_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	4	ue(v)
else if(remapping_of_pic_nums_idc == 2)		
long_term_pic_idx	4	ue(v)
} while(remapping_of_pic_nums_idc != 3)		
}		
}		
}		

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

7.3.3.2 Prediction weight table syntax

pred_weight_table( ) {	C	Descriptor
luma_log_weight_denom	4	ue(v)
chroma_log_weight_denom	4	ue(v)
for(i =0; i <= num_ref_idx_l0_active_minus1 ; i++) {		
luma_weight_l0_flag	4	u(1)
if(luma_weight_l0_flag) {		
luma_weight_l0[i]	4	se(v)
luma_offset_l0[i]	4	se(v)
}		
chroma_weight_l0_flag	4	u(1)
if(chroma_weight_l0_flag)		
for(j =0; j < 2; j++) {		
chroma_weight_l0[i][j]	4	se(v)
chroma_offset_l0[i][j]	4	se(v)
}		
}		
if(slice_type == B) {		
for(i = 0; i<= num_ref_idx_l1_active_minus1 ; i++) {		
luma_weight_l1_flag	4	u(1)
if(luma_weight_l1_flag) {		
luma_weight_l1[i]	4	se(v)
luma_offset_l1[i]	4	se(v)
}		
chroma_weight_l1_flag	4	u(1)
if(chroma_weight_l1_flag)		
for(j = 0; j < 2; j++) {		
chroma_weight_l1[i][j]	4	se(v)
chroma_offset_l1[i][j]	4	se(v)
}		
}		
}		
}		

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

7.3.3.3 Decoded picture buffer management syntax

dec_pic_buffer_management( ) {	C	Descriptor
adaptive_ref_pic_buffering_mode_flag	4 | 7	u(1)
if(adaptive_ref_pic_buffering_mode_flag == 1)		
do {		
memory_management_control_operation	4 | 7	ue(v)
if(memory_management_control_operation == 1 || memory_management_control_operation == 3)		
difference_of_pic_nums_minus1	4 | 7	ue(v)
if(memory_management_control_operation == 2 || memory_management_control_operation == 3)		
long_term_pic_idx	4 | 7	ue(v)
if(memory_management_control_operation == 4)		
max_long_term_pic_idx_plus1	4 | 7	ue(v)
} while(memory_management_control_operation != 0 && memory_management_control_operation < 5)		
}		

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

7.3.4 Slice data syntax

slice_data( ) {	C	Descriptor
if(entropy_code_mode == 1)		
while(!byte_aligned( ))		
cabac_alignment_bit	4	f(1)
if(mb_frame_field_adapative_flag && !field_pic_flag)		
MbAddress = first_mb_in_slice << 1		
else		
MbAddress = first_mb_in_slice		
MoreDataFlag = 1		
PrevMbSkipped = 0		
do {		
if(slice_type != I && slice_type != SI)		
if(entropy_coding_mode == 0) {		
mb_skip_run	4	ue(v)
if(mb_skip_run > 0)		
PrevMbSkipped = 1		
else		
PrevMbSkipped = 0		
for(i=0; i<mb_skip_run; i++)		
MbAddress = next_mb_addres s(MbAddress)		
MoreDataFlag = more_rbsp_data( )		
} else {		
mb_skip_flag	4	ae(v)
MoreDataFlag = !mb_skip_flag		
}		
if(MoreDataFlag) {		
if(mb_adaptive_frame_field_flag && !field_pic_flag) {		
if(MbAddress % 2 == 0 || (MbAddress % 2 == 1 && PrevMbSkipped)		
mb_field_decoding_flag	4	u(1) | ae(v)
} else		
mb_field_decoding_flag = 0		
macroblock_layer( )	4 | 5 | 6	
}		
MbAddress = next_mb_address(MbAddress)		
if(entropy_coding_mode == 0)		
MoreDataFlag = more_rbsp_data( )		
else {		
if(slice_type != I && slice_type != SI)		
PrevMbSkipped = mb_skip_flag		
if(mb_adaptive_frame_field_flag && !field_pic_flag && MbAddress % 2 != 0)		
MoreDataFlag = 1		
else {		
end_of_slice_flag	4	ae(v)
MoreDataFlag = !end_of_slice_flag		
}		

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

}		
} while(MoreDataFlag)		
}		

7.3.5 Macroblock layer syntax

macroblock_layer( ) {	C	Descriptor
mb_type	4	ue(v) | ae(v)
if(mb_type == PCM) {		
while(!byte_aligned())		
pcm_alignment_bit	4	f(1)
for(i = 0; i < 384; i++)		
pcm_byte	4	u(8)
} else {		
if((mb_part_pred_mode(mb_type, 0) != Intra_4x4 || mb_part_pred_mode(mb_type, 0) != Intra_16x16) && num_mb_part(mb_type) == 4)		
sub_mb_pred(mb_type)	4	
else		
mb_pred(mb_type)	4	
if(mb_part_pred_mode(mb_type, 0) != Intra_16x16)		
coded_block_pattern	4	ue(v) | ae(v)
if(coded_block_pattern > 0) {		
mb_qp_delta	4	se(v) | ae(v)
residual( )	5 | 6	
}		
}		
}		

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

7.3.5.1 Macroblock prediction syntax

mb_pred(mb_type) {	C	Descriptor
if(mb_part_pred_mode(mb_type, 0) == Intra_4x4 || mb_part_pred_mode(mb_type, 0) == Intra_16x16) {		
if(mb_part_pred_mode(mb_type, 0) == Intra_4x4)		
for(4x4LumaBlkIdx=0, 4x4LumaBlkIdx<16; 4x4LumaBlkIdx++) {		
prev_intra4x4_pred_mode_flag[4x4LumaBlkIdx **]**	4	u(1) | ae(v)
if(!prev_intra4x4_pred_mode_flag [4x4LumaBlkIdx])		
rem_intra4x4_pred_mode[4x4LumaBlkIdx **]**	4	u(3) | ae(v)
}		
intra_chroma_pred_mode	4	ue(v) | ae(v)
} else if(mb_part_pred_mode(mb_type, 0) != Direct) {		
for(MbPartIdx = 0; MbPartIdx < num_mb_part(mb_type); MbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 || mb_field_decoding_flag == 1) && mb_part_pred_mode(mb_type, MbPartIdx) != Pred_L1)		
ref_idx_l0[MbPartIdx **]**	4	te(v) | ae(v)
for(MbPartIdx = 0; MbPartIdx < num_mb_part(mb_type); MbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 || mb_field_decoding_flag == 1) && mb_part_pred_mode(mb_type, MbPartIdx) != Pred_L0)		
ref_idx_l1[MbPartIdx **]**	4	te(v) | ae(v)
for(MbPartIdx = 0; MbPartIdx < num_mb_part(mb_type); MbPartIdx++)		
if(mb_part_pred_mode (mb_type, MbPartIdx) != Pred_L1)		
for(CompIdx = 0; CompIdx < 2; CompIdx++)		
mvd_l0[0 **][** MbPartIdx **][** CompIdx **]**	4	se(v) | ae(v)
for(MbPartIdx = 0; MbPartIdx < num_mb_part(mb_type); MbPartIdx++)		
if(mb_part_pred_mode(mb_type, MbPartIdx) != Pred_L0)		
for(CompIdx = 0; CompIdx < 2; CompIdx++)		
mvd_l1[0 **][** MbPartIdx **][** CompIdx **]**	4	se(v) | ae(v)
}		
}		

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

7.3.5.2 Sub-macroblock prediction syntax

sub_mb_pred(mb_type) {	C	Descriptor
for(SubMbIdx = 0; SubMbIdx < 4; SubMbIdx++)		
sub_mb_type[SubMbIdx]	4	ue(v) | ae(v)
for(SubMbIdx = 0; SubMbIdx < 4; SubMbIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 || mb_field_decoding_flag = = 1) && mb_type != P_8x8ref0 && sub_mb_type[SubMbIdx] != B_Direct_8x8 && sub_mb_pred_mode(sub_mb_type[SubMbIdx]) != Pred_L1)		
ref_idx_l0[SubMbIdx]	4	te(v) | ae(v)
for(SubMbIdx = 0; SubMbIdx < 4; SubMbIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 || mb_field_decoding_flag = = 1) && sub_mb_type[SubMbIdx] != B_Direct_8x8 && sub_mb_pred_mode(sub_mb_type[SubMbIdx]) != Pred_L0)		
ref_idx_l1[SubMbIdx]	4	te(v) | ae(v)
for(SubMbIdx = 0; SubMbIdx < 4; SubMbIdx+ +)		
if(sub_mb_type[SubMbIdx] != B_Direct_8x8 && sub_mb_pred_mode(sub_mb_type[SubMbIdx]) != Pred_L1)		
for(SubMbPartIdx = 0; SubMbPartIdx < num_sub_mb_part(sub_mb_type [SubMbIdx]); SubMbPartIdx++)		
for(CompIdx = 0; CompIdx < 2; CompIdx++)		
mvd_l0[SubMbIdx][SubMbPartIdx][CompIdx]	4	se(v) | ae(v)
for(SubMbIdx = 0; SubMbIdx < 4; SubMbIdx++)		
if(sub_mb_type[SubMbIdx] != B_Direct_8x8 && sub_mb_pred_mode(sub_mb_type[SubMbIdx]) != Pred_L0)		
for(SubMbPartIdx = 0; SubMbPartIdx < num_sub_mb_part(sub_mb_type[SubMbIdx]); SubMbPartIdx++)		
for(CompIdx = 0; CompIdx < 2; CompIdx++)		
mvd_l1[SubMbIdx][SubMbPartIdx][CompIdx]	4	se(v) | ae(v)
}		

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

7.3.5.3 Residual data syntax

[Ed.Note: This subclause and the attached semantics and decoding process are broken …]

residual(mb_type) {	C	Descriptor
if(!entropy_coding_mode)		
residual_block = residual_block_cavlc( )	5 | 6	
else		
residual_block = residual_block_cabac( )	5 | 6	
if(mb_part_pred_mode(mb_type, 0) = = Intra_16x16)		
residual_4x4block(Intra16x16DCLevel, 16)	5	
for(i8x8 = 0; i8x8 < 4; i8x8++) /* each luma 8x8 block */		
for(i4x4 = 0; i4x4 < 4; i4x4++) /* each 4x4 sub-block of block */		
if(coded_block_pattern & (1 << i8x8)) [Ed.Note: coded_block_pattern is not parsed for Intra_16x16]		
if(mb_part_pred_mode(mb_type, 0) = = Intra_16x16)		
residual_block(Intra16x16ACLevel[i8x8 * 4 + i4x4], 15)	5	
else		
residual_block(LumaResidualLevel[i8x8 * 4 + i4x4], 16)	5 | 6	
if(coded_block_pattern & 0x30) [Ed.Note: coded_block_pattern is not parsed for Intra_16x16] /* chroma DC residual present */		
for(CbCrIdx = 0; CbCrIdx < 2; CbCrIdx++)		
residual_block(ChromaDCLevel[CbCrIdx], 4))	5 | 6	
if(coded_block_pattern & 0x20) /* chroma AC residual present */		
for(CbCrIdx = 0; CbCrIdx < 2; CbCrIdx++)		
for(i4x4 = 0; i4x4 < 4; i4x4++)		
residual_block(ChromaACLevel[CbCrIdx][i4x4], 15)	5 | 6	
}		

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

7.3.5.3.1 Residual 4x4 block CAVLC syntax

residual_block_cavlc(Level, MaxNumCoeff) {	C	Descriptor
coeff_token	5 | 6	ce(v)
NumCoeffs = total_coeff(coeff_token) - trailing_ones(coeff_token)		
if(trailing_ones(coeff_token) > 0)		
for(i = trailing_ones(coeff_token) -1; i >= 0; i--)		
trailing_ones_sign[i]	5 | 6	u(1)
if(total_coeff(coeff_token) > 0) {		
for(i = NumCoeffs-1; i >= 0; i--)		
coeff_level[i]	5 | 6	ce(v)
if(total_coeff(coeff_token) < MaxNumCoeff) {		
total_zeros	5 | 6	ce(v)
i = total_coeff(coeff_token) - 1		
ZerosLeft = total_zeros		
if(i>0 && ZerosLeft > 0) {		
do {		
run_before[i]	5 | 6	ce(v)
ZerosLeft -= run_before[i]		
i--		
} while(ZerosLeft > 0 && i > 0)		
run_before[i] = ZerosLeft		
}		
}		
}		
if(trailing_ones(coeff_token) > 0)		
for(i = 0 ; i < trailing_ones(coeff_token); i++)		
coeff_level[NumCoeffs + i + 1] = (1 – 2 * trailing_ones_sign[i]) [Ed.Note: is this right ? Further, no assignments to syntax elements: change]		
CoeffNum = -1		
for (i = 0; i < total_coeff(coeff_token); i ++) {		
CoeffNum += run_before[i] + 1		
Level[CoeffNum] = coeff_level[i]		
}		
}		

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

7.3.5.3.2 Residual 4x4 block CABAC syntax

residual_block_cabac(Level, MaxNumCoeff) {	C	Descriptor
coded_block_flag	5 | 6	ae(v)
if(coded_block_flag) {		
NumCoeff = MaxNumCoeff		
i = 0		
do {		
significant_coeff_flag[i]	5 | 6	ae(v)
if(significant_coeff_flag[i]) {		
last_significant_coeff_flag[i]	5 | 6	ae(v)
if(last_significant_coeff_flag[i]) {		
NumCoeff = i + 1		
for(j = NumCoeff; j < MaxNumCoeff; j++)		
Level[j] = 0		
}		
}		
i++		
} while(i < NumCoeff-1)		
coeff_absolute_value_minus1[NumCoeff-1]	5 | 6	ae(v)
coeff_sign_flag[NumCoeff-1]	5 | 6	ae(v)
Level[NumCoeff-1] = coeff_absolute_value_minus1[NumCoeff-1] * (1 – 2 * coeff_sign_flag[NumCoeff-1])		
for(i = NumCoeff-2; i >= 0; i--) {		
if(significant_coeff_flag[i]) {		
coeff_absolute_value_minus1[i]	5 | 6	ae(v)
coeff_sign_flag[i]	5 | 6	ae(v)
Level[i] = coeff_absolute_value_minus1[i] * (1 – 2 * coeff_sign_flag[i])		
} else		
Level[i] = 0		
}		
} else		
for(i = 0; i < MaxNumCoeff; i++) {		
Level[i] = 0		
}		

7.4 Semantics

7.4.1 NAL unit semantics

NOTE - The Video Coding Layer (VCL) is specified to efficiently represent the content of the video data. The Network Abstraction Layer (NAL) is specified to format that data and provide header information in a manner appropriate for conveyance by the transport layers or storage media. All data are contained in NAL units, each of which contains an integer number of bytes. A NAL unit specifies a generic format for use in both packet-oriented and bitstream systems. The format of NAL units for both packet-oriented transport and bitstream is identical except that each NAL unit can be preceded by a start code prefix in a bitstream-oriented transport layer.

NumBytesInNALunit specifies the size of the NAL unit in bytes. This value is required for decoding of the NAL unit and shall be conveyed by external means. Framing of NAL units is necessary to enable inference of NumBytesInNALunit. Such framing into a byte stream format is specified in Annex B and other methods for framing may be specified outside of this Recommendation | International Standard.

forbidden_bit shall be equal to 0.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

NOTE - The forbidden_bit may be used by external specifications to signal potentially corrupt NAL units for enhanced-capability uses not specified in the Recommendation | International Standard.

nal_ref_idc equal to 0 indicates that the content of the NAL unit belongs to a picture that is not marked for use as a reference picture in the decoded picture buffer, or contains SEI data or filler data. nal_ref_idc shall not be equal to 0 for sequence parameter set or picture parameter set NAL units. If nal_ref_idc is equal to 0 for one slice or slice data partition NAL unit of a particular picture, it shall be equal to 0 for all slice and slice data partition NAL units of the picture. nal_ref_idc greater than 0 indicates that the conten t of the NAL unit belongs to a decoded picture that is stored and marked for use as a reference picture in the decoded picture buffer.

NOTE - In addition to signalling whether a picture is marked for use as a reference picture, external specifications may use nal_ref_idc to indicate the relative transport priority of the NAL unit in a manner not specified in this Recommendation | International Standard. The value 0 should be used to signal the lowest transport priority and the priority should grow in ascending order of nal_ref_idc values.

nal_unit_type indicates the type of RBSP data structure contained in the NAL unit according to the types specified in Table 7-1. VCL NAL units are defined as those NAL units having nal_unit_type e qual to 1, 2, 3, 4, 5, or 10.

Table 7-1 – NAL unit type codes

nal_unit_type	Content of NAL unit and RBSP syntax structure	C
0	For external use	
1	Coded slice slice_layer_no_partitioning_rbsp( )	4, 5, 6
2	Coded slice data partition A slice_data_partition_a_layer_rbsp( )	4
3	Coded slice data partition B slice_data_partition_b_layer_rbsp( )	5
4	Coded slice data partition C slice_data_partition_c_layer_rbsp( )	6
5	Coded slice of an IDR picture slice_layer_no_partitioning_rbsp( )	4, 5
6	Supplemental enhancement information (SEI) sei_rbsp( )	7
7	Sequence parameter set (SPS) seq_parameter_set_rbsp( )	0
8	Picture parameter set (PPS) pic_parameter_set_rbsp( )	1
9	Picture delimiter (PD) pic_delimiter_rbsp( )	8
10	Filler data (FD) filler_data_rbsp( )	9
11 - 23	Reserved	
24 - 31	For external use	

[Ed.Note: This paragraph should be revised and possibly moved?] An instantaneous decoder refresh picture (IDR picture) implies that all pictures in the multi-picture buffer are marked as non-reference pictures. Moreover, the maximum long-term index is reset to zero. An IDR picture contains only I or SI slices, and nal_unit_type equal to 5 (IDR) shall be used for all slices of an IDR picture. nal_ref_idc shall be no t be equal to 0 for IDR NAL units. An IDR picture shall not immediately follow another IDR picture in decoding order.

The decoding process for NAL unit types 0 and 24-31 is not specified in this Recommendation | International Standard.

Note – NAL unit types 0 and 24-31 may be used as defined externally, for purposes such as NAL unit aggregation/fragmentation.

Decoders shall ignore (remove from the bitstream and discard) the contents of all NAL units which use reserved values of nal_unit_type.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

rbsp[i] a raw byte sequence payload is specified as an ordered sequence of bytes.

The RBSP contains an SODB in the following form:

a) If the SODB is null, the RBSP is also null.

b) Otherwise, the RBSP contains the SODB in the following form:

1) The first byte of the RBSP contains the (most significant, left-most) eight bits of the SODB; the next byte of the RBSP shall contain the next eight bits of the SODB, etc.; until fewer than eight bits of the SODB remain.

2) rbsp_trailing bits( ) are present after the SODB as follows:

i) The first (most significant, left-most) bits of the final RBSP byte contains the remaining bits of the SODB, if any,

ii) The next bit consists of a single rbsp_stop_bit equal to 1, and

iii) If the rbsp_stop_bit is not the last bit of a byte-aligned byte, one or more rbsp_alignment_bit is present to result in byte alignment.

3) One or more cabac_zero_word 16-bit syntax elements equal to 0x0000 may be present after the rbsp_trailing_bits() at the end of the RBSP.

Syntax structures having these RBSP properties are denoted in the syntax tables using an "_rbsp" suffix. These structures shall be carried within NAL units as the content of the rbsp[i] data bytes. The association of the RBSP syntax structures to the NAL units shall be as specified in Table 7-1.

If the boundaries of the RBSP are known, the decoder can extract the SODB from the RBSP by concatenating the bits of the bytes of the RBSP and discarding the rbsp_stop_bit, which is the last (least significant, right-most) bit equal to 1, and discarding any following (less significant, farther to the right) bits that follow it, which are equal to 0. The data necessary for the decoding process is contained in the SODB part of the RBSP.

emulation_prevention_byte is a byte equal to 0x03. If an emulation_prevention_byte is present in the NAL unit, it shall be discarded.

The last byte of the NAL unit shall not be equal to 0x00.

Within the NAL unit, the following three-byte sequences shall not occur at any byte-aligned position:

- 0x000000
- 0x000001
- 0x000002

The use of the byte-aligned 3-byte sequence 0x000002 within the NAL unit is reserved for future use by ITU-T | ISO/IEC. [Ed. Note: What is the decoder supposed to do if it sees that value in a byte stream? My guess is that it should choke, so maybe we don't need to state that we reserve that byte string.]

Within the NAL unit, any four-byte sequence which starts with 0x000003 other than the following sequences shall not occur at any byte-aligned position:

- 0x00000300
- 0x00000301
- 0x00000302
- 0x00000303

7.4.1.1 Example encoder procedure (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

The form of encapsulation of an SODB within an RBSP and the use of the emulation_prevention_byte for encapsulation of an RBSP within a NAL unit is specified for the following purposes:

- to prevent the emulation of start codes within NAL units while allowing any arbitrary SODB to be represented within a NAL unit,
- to enable identification of the end of the SODB within the NAL unit by searching the RBSP for the rbsp_stop_bit starting at the end of the RBSP, and
- to enable a NAL unit to have a size larger than that of the SODB under some circumstances (using one or more cabac_zero_word).

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

The encoder can produce a NAL unit from an RBSP by the following procedure:

The RBSP data is searched for byte-aligned bits of the following binary patterns:

'00000000 00000000 000000xx' (where xx represents any 2 bit pattern: 00, 01, 10 or 11),

and a byte equal to 0x03 is inserted to replace these bit patterns with the patterns

'00000000 00000000 00000011 000000xx',

and finally, if the last byte of the RBSP data is equal to 0x00 (which can only occur if the RBSP ends in a cabac_zero_word), a final byte equal to 0x03 is appended to the end of the data.

The resulting sequence of bytes is then prefixed with the first byte of the NAL unit containing the indication of the type of RBSP data structure it contains. This results in the construction of the entire NAL unit.

This process can allow any SODB to be represented in a NAL unit while ensuring that

- no byte-aligned start code prefix is emulated within the NAL unit, and
- no sequence of 8 zero-valued bits followed by a start code prefix, regardless of byte-alignment, is emulated within the NAL unit.

7.4.2 Raw byte sequence payloads and RBSP trailing bits semantics

7.4.2.1 Sequence parameter set RBSP semantics

[Ed.Note: Move the following pars. to the NAL unit order semantics clause. Change the meaning of "active"] A sequence parameter set is called an active sequence parameter set when an IDR NAL unit refers to it. The parameters of an active sequence parameter set shall be replaced only when an IDR NAL unit refers to a different sequence parameter set.

A picture parameter set includes the parameters that remain unchanged within a coded picture. Every picture parameter set shall refer to the active sequence parameter set. A decoded picture parameter set is an active picture parameter set when the first slice NAL unit or first slice data partitionA NAL unit of a coded picture refers to it. The picture parameters of an active picture parameter set shall be replaced only when the first slice NAL unit or slice data partition A NAL unit of a picture refers to a different picture parameter set.

NOTE - The sequence and picture parameter set mechanism decouples the transmission of infrequently changing information from the transmission of coded macroblock data. The sequence parameter set, the picture parameter set, and the slice header contain all the parameters needed to decode the slice data. It is recommended to convey sequence and picture parameter sets out-of-band using a reliable transport mechanism. However, if an application requires a self-contained bitstream, in-band parameter set information units may be used. In error-prone transmission environments, in-band sequence and picture parameter set information units should be protected in a way that assures their successful reception. Synchronization between in-band and out-of-band transmission of the sequence and picture parameter set information is outside of the scope of this Recommendation | International Standard.

profile_idc and **level_idc** indicate profile and level as specified in Annex A. The value of profile_idc shall be in the range from 65 to 90, inclusive. The value of level_idc shall be in the range from 0 to 109, inclusive.

seq_parameter_set_id identifies the sequence parameter set to be referred. The value of seq_parameter_set_id shall be in the range of 0 to 31, inclusive.

log2_max_frame_num_minus4 specifies the MaxFrameNum used in frame number related arithmetic as follows:

$$\text{MaxFrameNum} = 2^{(\text{log2\_max\_frame\_num\_minus4} + 4)} \quad (7\text{-}1)$$

The value of log2_max_frame_num_minus4 shall be in the range of 0 to 12, inclusive.

pic_order_cnt_type equal to 0 or 1 indicates the method to code picture order count (see subclause 8.2.2). If pic_order_cnt_type equals 2, no picture order count information is coded , and there shall be no B slices and no two consecutive non-reference pictures in decoding order in the sequence. The value of pic_order_cnt_type shall be in the range of 0 to 2, inclusive.

delta_pic_order_always_zero_flag equal to 1 indicates that delta_pic_order_cnt[0] and delta_pic_order_cnt[1] are always 0 and are not coded. delta_pic_order_always_zero_flag equal to 0 indicates that delta_pic_order_cnt[0] is coded and delta_pic_order_cnt[1] may be coded.

offset_for_non_ref_pic indicates an expected picture order count difference of a non-reference picture compared to the expected picture order count of the most recently decoded reference picture having a frame_num one less (modulo MaxFrameNum) than that of the non-reference picture. The value of offset_for_non_ref_pic shall be in the range from -2^{31} to $2^{31}-1$, inclusive.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

offset_for_top_to_bottom_field indicates an expected picture order count difference from the top field to the bottom field of a frame picture. The value of offset_for_top_to_botto m_field shall be in the range from -2^{31} to $2^{31}-1$, inclusive.

num_ref_frames_in_pic_order_cnt_cycle signals the number of frame numbers in a picture order count cycle. A picture order count cycle is a repetitive pattern of picture order count differences, e ach of which corresponds to a frame_num increment of one. The value of num_ref_frames_in_pic_order_cnt_cycle shall be in the range from 0 to 255.

offset_for_ref_frame indicates an expected difference of picture order count corresponding to a frame_num incr ement of one. The notation offset_for_ref_frame [i] indicates the offset_for_ref_frame corresponding to index 'i' in the picture order count cycle. The value of offset_for_ref_frame shall be in the range from -2^{31} to $2^{31}-1$, inclusive.

num_ref_frames specifies the maximum total number of short - and long-term reference pictures in the decoded picture buffer. num_ref_frames also determines the size of the Sliding Window operation as specified in subclause 8.2.7.2. Min value is 0. [Ed.Note: really? clash with successive IDRs rule.]. Max value is specified in Annex A.

required_frame_num_update_behaviour_flag equal specifies the decoder behaviour in case of missing frame numbers as specified in subclause 8.2.6.2.

frame_width_in_mbs_minus1 and **frame_height_in_mbs_minus1** specify the width and height, respectively, of the decoded frame in units of macroblocks. The frame width and height in units of macroblocks is computed by adding 1 to the decoded values of each of these parameters. The allowed range of values for these syntax elements is specified in Annex A. If mb_adaptive_frame_field_flag is equal to 1 or field_pic_flag is equal to 1, frame_height_in_mbs_minus1+1 shall be an integer multiple of 2. [Ed.Note: ch ange to frame_height_in_mb_pairs_minus1]

The picture width in units of macroblocks, PicWidthInMbs, is equal to frame_width_in_mbs_minus1 + 1. If a picture is a coded frame, the picture height in units of macroblocks, PicHeightInMbs, shall be equal to frame_height_in_mbs_minus1 + 1. If the picture is a coded field, PicHeightInMbs shall be calculated according to:

$$\text{PicHeightInMbs} = (\text{frame\_height\_in\_mbs\_minus1} + 1) / 2 \quad (7\text{-}2)$$

The maximum macroblock address, MaxMbAdd ress, shall be calculated according to:

$$\text{MaxMbAddress} = \text{PicWidthInMbs} * \text{PicHeightInMbs} - 1 \quad (7\text{-}3)$$

frame_coding_only_flag equal to 0 specifies that a coded picture can be a coded field or a coded frame. frame_coding_only_flag equal to 1 specifies that every coded picture is coded as a frame.

mb_frame_field_adaptive_flag equal to 0 indicates no switching between frame and field decoding mode at the macroblock layer, whereas equal to 1 indicates the possible use of s witching between frame and field decoding mode at the macroblock layer. If a picture is a coded field (if field_pic_flag is 1), then the value of mb_frame_field_adaptive_flag shall be ignored.

direct_temporal_constrained_flag specifies the method used in t he decoding process to determine the prediction values of temporal direct prediction. Subclause 8.4.1.4.2 describes the temporal direct prediction and how the motion parameters are calculated based on the value of direct_tempora l_constrained_flag. If this flag is equal to 1 then 4x4, 4x8, and 8x4 partition sizes are not allowed in temporal direct mode macroblocks.

vui_seq_parameters_present_flag equal to 0 specifies that default parameter values for the VUI sequence parameters shall be applied. Equal to 1 specifies that VUI sequence parameters are coded next. VUI sequence parameters are specified in Annex E.

7.4.2.2 Picture parameter set RBSP semantics

pic_parameter_set_id identifies the picture parameter set to be referred. The value of pic_parameter_set_id shall be in the range of 0 to 127, inclusive.

seq_parameter_set_id identifies the sequence parameter set that is used with this picture parameter set. The value of seq_parameter_set_id shall be in the range of 0 to 31, inclusive.

entropy_coding_mode selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables. If entropy_coding_mode is equal to 0 the method indicated by the left descriptor is applied (exp-Golomb coded, see subclause 9.1 or CAVLC, see subclause 9.2), whereas if entropy_coding_mode is equal to 1 the method indicated by the right descriptor is applied (CABAC, see subclause 9.3). The value of entropy_coding_mode shall be in the range of 0 to 1, inclusive. [Ed.Note: turn this into a flag, which it is at the end of the day]

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

pic_order_present_flag equal to 0 indicates that only delta_pic_order_cnt[0] (without delta_pic_order_cnt[1]) is sent in the slice_header, while equal to 1 indicates that both delta_pic_order_cnt[0] and delta_pic_order_cnt[1] are sent in the slice_header.

num_slice_groups_minus1 the number of slice groups is equal to num_slice_groups_minus1 + 1. If num_slice_groups_minus1 is zero, all slices of the picture belong to the same slice group. Allowed range is specified in Annex A.

NOTE – One slice group means that no flexible macroblock ordering is applied. If num_slice_groups_minus1 is greater than zero, flexible macroblock ordering is in use.

mb_slice_group_map_type indicates how the mapping of macroblocks to slice groups is coded. The value of mb_slice_group_map_type shall be in the range of 0 to 6, inclusive.

mb_slice_group_map_type 0 is used to indicate interleaved slices.

mb_slice_group_map_type 1 is used to indicate a dispersed macroblock mapping to slice groups.

mb_slice_group_map_type 2 is used to indicate one or more "foreground" slice groups and a "leftover" slice group.

mb_slice_group_map_types 3, 4 and 5 are used to indicate changing slice groups. num_slice_groups_minus1 shall be 1, when mb_slice_group_map_type is equal to 3, 4 or 5.

mb_slice_group_map_type 6 is used to explicitly assign a slice group to each macroblock or macroblock pair location in raster scan order.

If macroblock-adaptive frame/field coding is not in use, the mapping of macroblocks to slice groups operates in units of macroblocks. If macroblock-adaptive frame/field coding is in use, the macroblock allocation map works in units of macroblock pairs.

If mb_slice_group_map_type is 0, the run_length syntax element follows for each slice group.

run_length[i] indicates the number of consecutive macroblocks or macroblock pairs that are assigned to the ith slice group in macroblock or macroblock pair raster scan. After the macroblocks or macroblock pairs of the last slice group have been assigned, the process begins again from the first slice group. The process ends when all the macroblocks of a picture have been assigned. The value of run_length shall be in the range from 0 to MaxMbAddress, inclusive. [Ed.Note: broken, since you don't know how many mbs are in a picture if you don't know whether it is a frame or field at this point. Not entirely broken (you can parse the pic param set once you have the slice header), but undesirable.]

NOTE - Example: To signal macroblock row interleaving in a QCIF picture (where all even numbered macroblocks are in slice group 0, and all odd numbered macroblocks are in slice group 1), the number of slice groups is two and run_length is 11 for both slice groups.

If mb_slice_group_map_type is equal to 1, the macroblock or macroblock pair mapping to slice groups is formed using the following formula, where n is the number of columns in the picture (in macroblocks) and p is the number of slice groups to be coded. Specifically, macroblock or macroblock pair position x is assigned to slice group S according to:

$$S(x, n, p) = ((x \% n) + (((x / n) * p) / 2)) \% p \qquad (7\text{-}4)$$

[Ed. Note: Pick a longer string for the name and move this to subclause 5.6]

If mb_slice_group_map_type is equal to 2, **top_left_mb[i]** and **bottom_right_mb[i]** are specified for the ith slice group except for the last one in ascending order of slice group IDs starting from 0.

top_left_mb[i] and **bottom_right_mb[i]** specify the top-left and bottom-right corners of a rectangle, respectively. top_left_mb[i] and bottom_right_mb[i] are indicated as macroblock addresses or macroblock pair addresses. For each rectangle i, top_left_mb[i] shall be smaller than bottom_right_mb[i]. The rectangles may overlap. Slice groups are allocated in ascending order of slice group IDs. Slice group 0 contains the macroblocks that are within the indicated rectangle top_left_mb[0] and bottom_right_mb[0]. A slice group having slice group ID greater than 0 contains the macroblocks that are within the indicated rectangle for that slice group and that do not belong to any slice group having a smaller slice group ID, i.e. the slice group with the largest slice group ID contains the macroblocks that are not covered by the other slice groups. Each slice groups shall contain at least one macroblock. The size of the top_left_mb and bottom_right_mb parameters shall be NumBitsInMbAddress. [Ed. Note (Karsten): this is broken. We cannot calculate NumBitsInMbAddress with the current definition at this point. (see Note above). Not entirely broken (you can parse the pic param set once you have the slice header), but undesirable.]

The NumBitsInMbAddress indicates the number of bits used to represent a macroblock address with a fixed-length unsigned integer.

If macroblock-adaptive frame/field coding is not in use for a picture (if mb_adaptive_frame_field_flag is equal to 0 or if field_pic_flag is equal to 1), then NumBitsInMbAddress shall be calculated according to:

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

$$NumBitsInMbAddress = Ceil(Log2(MaxMbAddress + 1)) \quad (7\text{-}5)$$

If macroblock-adaptive frame/field coding is in use for a picture (if mb_adaptive_frame_field_flag is equal to 1 and field_pic_flag is equal to 0), NumBitsInMbAddress shall be calculated according to:

$$NumBitsInMbAddress = Ceil(Log2(MaxMbAddress + 1) - 1) \quad (7\text{-}6)$$

slice_group_change_direction_flag is used with mb_slice_group_map_type to indicate the refined type of the mb_slice_group_map according to Table 7-2 when mb_slice_group_map_type is 3, 4, or 5. The macroblock to slice group map is generated each time the decoder starts decoding of a new picture as described in subclause 8.2.4.

Table 7-2– Refined macroblock to slice group map type

mb_slice_group_map_type	slice_group_change_direction_flag	refined macroblock to slice group map type
4	0	Box-out clockwise
4	1	Box-out counter-clockwise
5	0	Raster scan
5	1	Reverse raster scan
6	0	Wipe right
6	1	Wipe left

slice_group_change_rate_minus1 is the minimum [Ed.Note: ?] non-zero number of macroblocks or macroblock pairs by which the size a slice group can change from one picture to the next. The SliceGroupChangeRate variable is specified as follows:

$$SliceGroupChangeRate = slice\_group\_change\_rate\_minus1 + 1 \quad (7\text{-}7)$$

The value of slice_group_change_rate_minus1 shall be in the range of 0 to MaxMbAddress – 1, inclusive.

If mb_slice_group_map_type is equal to 6, **slice_group_id** identifies a slice group of a macroblock or macroblock pair. The size of the slice_group_id parameter is Ceil(Log2(num_slice_groups_minus1)) bits. The value of slice_group_id shall be in the range of 0 to num_slice_groups_minus1, inclusive.

NOTE - slice_group_id is repeated as often as there are macroblocks in the picture.

num_ref_idx_l0_active_minus1 specifies the maximum reference picture index minus 1 in the reference list 0 that are used to decode the picture. The value of num_ref_idx_l0_active_minus1 is in the range of 0 to 14, inclusive.

num_ref_idx_l1_active_minus1 specifies the maximum reference picture index minus 1 in the reference list 1 that are used to decode the picture. The value of num_ref_idx_l1_active_minus1 is in the range of 0 to 14, inclusive.

weighted_pred_flag equal to 0 indicates that weighted prediction is not applied to P and SP slices. weighted_pred_flag equal to 1 indicates that weighted prediction is applied to P and SP slices.

weighted_bipred_explicit_flag equal to 0 indicates that explicit weighted prediction is not applied to B slices. weighted_bipred_explicit_flag equal to 1 indicates that explicit weighted prediction is applied to B slices. weighted_bipred_explicit_flag shall be equal to 0 if weighted_bipred_implicit_flag is one.

weighted_bipred_implicit_flag equal to 0 indicates that implicit weighted prediction is not applied to B slices. weighted_bipred_implicit_flag equal to 1 indicates that implicit weighted prediction is applied to B slices. weighted_bipred_implicit_flag shall be equal to 0 if weighted_bipred_explicit_flag is equal to 1.

pic_init_qp_minus26 specifies the initial value minus 26 of QP_Y for each slice. The initial value is modified at the slice layer when a non-zero value of slice_qp_delta is decoded, and is modified further when a non-zero value of mb_qp_delta is decoded at the macroblock layer. The value of pic_init_qp_minus26 shall be in the range of -26 to +25, inclusive.

pic_init_qs_minus26 specifies the initial value minus 26 of QS_Y for all macroblocks in SP or SI slices The initial value is modified at the slice layer when a non-zero value of slice_qs_delta is decoded. The value of pic_init_qs_minus26 shall be in the range of -26 to +25, inclusive.

chroma_qp_index_offset specifies the offset that shall be added to QP_Y and QS_Y for addressing the table of QP_C values. The value of chroma_qp_index_offset shall be in the range from -12 to +12, inclusive.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

deblocking_filter_parameters_present_flag specifies whether a set of parameters controlling the characteristics of the deblocking filter is indicated in the slice header.

constrained_intra_pred_flag equal to 0 indicates that intra prediction allows use of neighbouring non -intra macroblock data and decoded samples for the prediction of intra macroblocks, whereas constrained_intra_pred_flag equal to 1 indicates constrained intra prediction, where no intra prediction only uses data and decoded samples from I or SI macroblock types.

redundant_slice_flag indicates the presence of the redundant_pic_cnt parameter in all slice headers referencing the picture parameter set.

vui_pic_parameters_flag equal to zero specifies that default parameter values for the vui picture parameters shall be applied. [Ed.Note: mention Annex E]

7.4.2.3 Supplemental enhancement information RBSP semantics

Supplemental Enhancement Information (SEI) co ntains information that is not necessary to decode VCL data correctly but is helpful for decoding or presentation purposes.

7.4.2.3.1 Supplemental enhancement information message semantics

An SEI NAL unit contains one or more SEI messages. Each SEI message consists of SEI header and SEI payload. The type and size of the SEI payload are coded using an extensible syntax. The SEI payload size is indicated in bytes. SEI payload types are specified in Annex C.

The SEI payload may have an SEI payload header. For example, a payload beader may indicate to which picture the particular data belongs. The payload header shall be specified for each payload type separately.

An SEI message is associated with the next slice or slice data partition RBSP in decoding order. [Ed.Note: isn't it associated with a complete picture ? Will it remain active for the next pictures ?]

byte_ff is a byte equal to 0xFF identifying a need for a longer representation of the syntax structure it is used within.

last_payload_type_byte identifies the payload type of the last entry in an SEI message.

last_payload_size_byte identifies the size of the last entry in an SEI message.

7.4.2.4 Picture delimiter RBSP semantics

The picture delimiter may be used to signal the boundary between pictures, i.e., if present, it shall be inserted before the first [Ed.Note: Which is the first?] NAL unit of a picture in decoding order. There is no normative decoding process associated with the picture delimiter.

pic_type indicates which slice types are used in the following picture in decoding order. Table 7-3 shows the slice types that can occur in a picture with a given pic_type.

Table 7-3– Meaning of pic_type

pic_type	slice_type's allowed in picture
0	I
1	I, P
2	I, P, B
3	SI
4	SI, SP
5	I, SI
6	I, SI, P, SP,
7	I, SI, P, SP, B

7.4.2.5 Filler data RBSP semantics

The filler data RBSP contains bytes whose value shall be equal to 0xFF.

byte_ff is a byte equal to 0xFF.

7.4.2.6 Slice layer RBSP semantics

The slice layer RBSP consists of a slice header and slice data.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

7.4.2.7 Slice data partition RBSP semantics

7.4.2.7.1 Slice data partition A RBSP semantics

When slice data partitioning is in use, the coded data for a single slice is divided into three separate partitions. Partition A contains all symbols of Category 4.

NOTE - Category 4 consists of the header symbols of all coded macroblocks.

slice_id each slice of a picture is associated a unique slice identifier within the picture, when redundant_slice_flag is equal to 0. If redundant_slice_flag is equal to 1, each slice has a unique slice identifier among the slices having the same redundant_pic_cnt. Within any primary or redundant coded picture, the first coded slice of the picture shall have identifier equal to 0 and the identifier shall be incremented by one per each coded slice in decoding order. The range of slice_id is specified in Annex A. [Ed.Note: not yet done in Annex A. If not specified otherwise, this will become 0 to MaxMbAddress.]

7.4.2.7.2 Slice data partition B RBSP semantics

When slice data partitioning is in use, the coded data for a single slice is divided into three separate partitions. Slice data partition B contains all symbols of Category 5.

NOTE - Category 5 consists of the coded block patterns and coefficients for I and SI macroblock types.

slice_id each slice of a picture is associated a unique slice identifier within the picture, when redundant_slice_flag is equal to 0. If redundant_slice_flag is equal to 1, each slice has a unique slice identifier among the slices having the same redundant_pic_cnt. Within any primary or redundant coded picture, the first coded slice of the picture shall have identifier equal to 0 and the identifier shall be incremented by one per each coded slice in decoding order. The range of slice_id is specified in Annex A. [Ed.Note: not yet done in Annex A. If not specified otherwise, this will become 0 to MaxMbAddress.]

redundant_pic_cnt is equal to 0 for slices and slice data partitions belonging to the primary coded picture. The redundant_pic_cnt is greater than 0 for redundant coded slices and slice data partitions with max value being 127.

7.4.2.7.3 Slice data partition C RBSP semantics

When slice data partitioning is in use, the coded data for a single slice is divided into three separate partitions. Slice dat a partition C contains all symbols of Category 6.

NOTE - Category 6 consists of the coded block patterns and coefficients for P and B macroblock types.

slice_id each slice of a picture is associated a unique slice identifier within the picture, when redund ant_slice_flag is equal to 0. If redundant_slice_flag is equal to 1, each slice has a unique slice identifier among the slices having the same redundant_pic_cnt. Within any primary or redundant coded picture, the first coded slice of the picture shall have identifier equal to 0 and the identifier shall be incremented by one per each coded slice in decoding order. The range of slice_id is specified in Annex A. [Ed.Note: not yet done in Annex A. If not specified otherwise, this will become 0 to MaxMbAddress.]

redundant_pic_cnt is equal to 0 for slices and slice data partitions belonging to the primary coded picture. The redundant_pic_cnt is greater than 0 for redundant coded slices and slice data partitions with max value being 127.

7.4.2.8 RBSP slice trailing bits semantics

cabac_zero_word is a byte-aligned string of two bytes equal to 0x0000. When entropy_coding_mode is equal to 1, the number of bins (excluding end_of_slice_flag) resulting from decoding the contents of the slice layer NAL unit shall not exceed 32*NumBytesInNALunit.

NOTE – The constraint on the maximum number of bins resulting from decoding the contents of the slice layer NAL unit can be met by inserting a number of cabac_zero_word syntax elements to increase the value of NumBytesInNALunit. Each cabac_zero_word is represented in the NAL unit by the three-byte sequence 0x000003 (as a result of the constraints on NAL unit contents that result in requiring inclusion of an emulation_prevention_byte for each cabac_zero_word).

7.4.2.9 RBSP trailing bits semantics

rbsp_stop_bit is a single bit equal to 1.

rbsp_alignment_bit is a single bit equal to 0.

7.4.3 Slice header semantics

pic_parameter_set_id indicates the picture parameter set in use. The value of pic_parameter_set_id shall be in the range of 0 to 127, inclusive.

frame_num identifies the frame for the picture. frame_num shall be incremented by 1 for each coded picture in decoding order, in modulo MaxFrameNum operation, relative to the frame_num of the last previous reference picture in

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

decoding order that has a different value of frame_num. An IDR picture shall have frame_num equal to 0. A frame picture shall not have the same frame_num as the reference picture that immediately precedes it in decoding order. A field picture shall not have the same frame_num as the reference picture that immediately precedes it in decoding order unless that picture is another field picture and has opposite parity. The decoding order of primary coded pictures shall be non-decreasing in frame number order. For non-reference pictures with the same frame number, the decoding order shall be non-decreasing in picture order count. The frame_num serves as a unique identifier for each reference frame in the decoded picture buffer. Therefore, a reference frame cannot be kept in the buffer aft er its frame_num has been used by another reference frame unless it has been assigned a long -term frame index as specified below. The frame_num of a reference frame to be added to the decoded picture buffer shall not be equal to the frame_num of any other among the short-term reference frames in the decoded picture buffer.

A picture including a memory_management_control_operation equal to 5 (reset all reference pictures) or 7 (reset all reference and non-reference pictures) shall have frame_num as describe d above, but after processing of the memory_management_control_operations, shall be considered to have frame_num equal to 0. frame_num is represented by log2_max_frame_num_minus4 + 4 bits in the bitstream.

field_pic_flag equal to 1 indicates that the pictu re is a coded field and equal to 0 indicates that the picture is a coded frame.

bottom_field_flag is only present when field_pic_flag is equal to 1. If bottom_field_flag is present and equal to 1, the picture is a coded bottom field. If bottom_field_flag is present and equal to 0, the picture is a coded top field.

first_mb_in_slice specifies the location of the first macroblock contained in this slice. The size of the first_mb_in_slice parameter is NumBitsInMbAddress. first_mb_in_slice shall be in the ran ge of 0 to MaxMbAddress, inclusive.

When macroblock-adaptive frame/field coding is not in use (when mb_adaptive_frame_field_flag is equal to 0 or field_pic_flag is equal to 1), first_mb_in_slice contains a macroblock address.

When macroblock-adaptive frame/field coding is in use (when mb_adaptive_frame_field_flag is equal to 1 and field_pic_flag is equal to 0), first_mb_in_slice contains a macroblock pair address and the number of macroblocks in the slice shall be an integer multiple of 2.

slice_type indicates the coding type of the slice according to Table 7-4.

Table 7-4 – Name association to slice_type

slice_type	Name of slice_type
0	P (P slice)
1	B (B slice)
2	I (I slice)
3	SP (SP slice)
4	SI (SI slice)

delta_pic_order_cnt[0] indicates the picture order count difference from the expected picture order count for the top field of a coded frame or for a coded field as described in subclause 8.2.2. If delta_pic_order_always_zero_flag is equal to 1, then delta_pic_order_cnt[0] is set equal to 0 and is not coded. The value of delta_pic_order_cnt[0] shall be in the range of -2^{31} to $2^{31}-1$, inclusive.

delta_pic_order_cnt[1] indicates the picture order count difference from the expected picture order count for the bottom field of a coded frame described in subclause 8.2.2. If delta_pic_order_always_zero_flag is equal to 1, then delta_pic_order_cnt[1] is set equal to 0 and is not coded. The value of delta_pic_order_cnt[0] shall be in the range of -2^{31} to $2^{31}-1$, inclusive.

redundant_pic_cnt is equal to 0 for slices and slice data partitions belonging to the primary coded picture. The redundant_pic_cnt is gre ater than 0 for redundant coded slices and slice data partitions with max value being 127.

NOTE - There should be no noticeable difference between the co-located areas of the decoded primary picture and any decoded redundant pictures.

Redundant slices and slice data partitions having the same value of redundant_pic_cnt belong to the same redundant picture. Decoded slices within the same redundant picture may not cover the entire picture area and shall not overlap. If the value of nal_ref_idc in a primary pi cture is equal to 0, the nal_ref_idc in corresponding redundant pictures shall be equal to 0. If the value of nal_ref_idc in a primary picture is greater than 0, the nal_ref_idc in corresponding redundant pictures shall be greater than 0.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

The slices of a redundant picture shall have the same values of field_pic_flag and bottom_field_flag as the corresponding primary picture. The frame_num of a redundant picture shall be the same as the frame_num in the primary picture, or both the redundant picture and the primary picture shall issue a reset memory management control operation (which resets the frame number to 0). The calculated picture order count of a redundant picture shall be the same as for the corresponding primary picture.

direct_spatial_mv_pred_flag specifies the method used in the decoding process to determine the prediction values direct prediction. If direct_spatial_mv_pred_flag is equal to 0, then direct mode motion parameters are calculated from the picture order count as described in subclause 8.4.1.4.2.3. If this flag is equal to 1, then direct mode motion parameters are calculated using the spatial motion vector prediction technique as described in subclause 8.4.1.4.2.2. [Ed.Note: semantics should be more specific]

num_ref_idx_active_override_flag equal to 0 indicates that the num_ref_idx_l0_active_minus1 and num_ref_idx_l1_active_minus1 specified in the referred picture parameter set are in effect. num_ref_idx_active_override_flag equal to 1 indicates that the num_ref_idx_l0_active_minus1 and num_ref_idx_l1_active_minus1 specified in the referred picture parameter set are overridden for the current slice only by the following values in the slice header.

num_ref_idx_l0_active_minus1 specifies the maximum reference picture index minus 1 in the reference picture list 0 that are used to decode the slice. The range of num_ref_idx_l0_active_minus1 is specified in Annex A. [Ed. Note: Not yet specified there. If not specified otherwise, it will be 0 to 14, inclusive.]

num_ref_idx_l1_active_minus1 specifies the maximum reference picture index minus 1 in the reference picture list 1 that are used to decode the slice. The range of num_ref_idx_l1_active_minus1 is specified in Annex A. [Ed. Note: Not yet specified there. If not specified otherwise, it will be 0 to 14, inclusive.]

non_ref_pic_reset_flag indicates how the non-reference pictures are treated for an IDR picture. A value of 1 indicates that the non-reference pictures are removed from the decoded picture buffer. A value of 0 indicates that the non-reference pictures remain in the decoded picture buffer.

slice_qp_delta specifies the initial value of QP_Y to be used for all the macroblocks in the slice until modified by the value of mb_qp_delta in the macroblock layer. The initial QP_Y parameter for the slice is computed as:

$$QP_Y = 26 + pic\_init\_qp\_minus26 + slice\_qp\_delta \quad (7\text{-}8)$$

The value of slice_qp_delta shall be limited such that QP_Y is in the range of 0 to 51, inclusive.

sp_for_switch_flag indicates the decoding process to be used to decode the SP slice. [Ed.Note: How about details? What further detail is necessary here?]

slice_qs_delta specifies the value of QS_Y for all the macroblocks in SP and SI slices. The QS_Y parameter for the slice is computed as:

$$QS_Y = 26 + pic\_init\_qs\_minus26 + slice\_qs\_delta \quad (7\text{-}9)$$

The value of slice_qs_delta shall be limited such that QS_Y is in the range of 0 to 51, inclusive. This value of QS_Y is used for the decoding of all macroblocks in the slice.

NOTE - This syntax element is signalled for SP and SI slices only.

disable_deblocking_filter_idc indicates whether the operation of the deblocking filter shall be disabled across some block edges of the slice and indicates on which edges the filtering is disabled. If disable_deblocking_filter_idc is not present in the slice header, the value of disable_deblocking_filter_idc shall be inferred to be equal to 0.

If disable_deblocking_filter_idc is equal to 0, the deblocking filter shall be applied across all edges controlled by the macroblocks within the current slice.

If disable_deblocking_filter_idc is equal to 1, the deblocking filter shall not be applied across any edges controlled by the macroblocks within the current slice.

If disable_deblocking_filter_idc is equal to 2, the deblocking filter shall be applied across all edges controlled by the macroblocks within the current slice, with the exception of the macroblock edges that are also slice boundaries.

The value of disable_deblocking_filter_idc shall be in the range of 0 to 2.

slice_alpha_c0_offset_div2 specifies the offset used in accessing the α and C0 deblocking filter tables for filtering operations controlled by the macroblocks within the slice. From this value, the offset that shall be applied when addressing these tables shall be computed as:

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

$$FilterOffsetA = slice\_alpha\_c0\_offset\_div2 << 1 \quad (7\text{-}10)$$

The value of slice_alpha_c0_offset_div2 shall be in the range of -6 to +6, inclusive. If slice_alpha_c0_offset_div2 is not present in the slice header the value of slice_alpha_c0_offset_div2 shall be inferred to be equal to 0.

slice_beta_offset_div2 specifies the offset used in accessing the β deblocking filter table for filtering operations controlled by the macroblocks within the slice. From this value, the offset that is applied when addressing the β table of the deblocking filter shall be computed as:

$$FilterOffsetB = slice\_beta\_offset\_div2 << 1 \quad (7\text{-}11)$$

The value of slice_beta_offset_div2 shall be in the range of -6 to +6, inclusive. If slice_beta_offset_div2 is not present in the slice header the value of slice_beta_offset_div2 shall be inferred to be equal to 0.

slice_group_change_cycle × SliceGroupChangeRate indicates the number of macroblocks in slice group 0. The value of slice_group_change_cycle is represented in the bitstream by the following number of bits

$$Ceil(Log2(Ceil((MaxMbAddress+1) \div SliceGroupChangeRate))).$$

The minimum value of slice_group_change_cycle is 0. The maximum value of slice_group_change_cycle is

$$Ceil((MaxMbAddress+1) \div SliceGroupChangeRate).$$

The value of pic_parameter_set_id, frame_num, field_pic_flag, bottom_field_flag, delta_pic_order_cnt[0], delta_pic_order_cnt[1], sp_for_switch_flag, and slice_group_change_cycle shall be the same in all slice headers of a redundant or primary coded picture. Moreover, the parameter values included in the dec_pic_buf_management( ) function shall remain unchanged in all slice headers of a redundant or primary coded picture.

7.4.3.1 Reference index reordering semantics

The syntax elements remapping_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_idx specify the change from the default reference index lists to the reference index lists used for decoding the slice.

ref_idx_reordering_flag_l0 equal to 1 indicates that the syntax elements remapping_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_idx are present for specifying the reference index list 0, while ref_idx_reordering_flag_l0 equal to 0 indicates that these syntax elements are not present.

ref_idx_reordering_flag_l1 equal to 1 indicates that the syntax elements remapping_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_idx are present for specifying the reference index list 1, while ref_idx_reordering_flag_l1 equal to 0 indicates that these syntax elements are not present.

remapping_of_pic_nums_idc together with abs_diff_pic_num_minus1 and long_term_pic_idx indicates which of the reference pictures are re-mapped. The values of remapping_of_pic_nums_idc are specified in Table 7-5. The number of signalling remapping_of_pic_nums_idc is limited to the num_ref_idx_l0_active_minus1 + 1. [Ed.Note: is this true with weighted prediction ? Yes, because num_ref_idx_active_minus1+1 is or should be the number of indexes active, not the number of pictures active.]

Table 7-5 – remapping_of_pic_nums_idc operations for re-mapping of reference pictures

remapping_of_pic_nums_idc	Re-mapping specified
0	abs_diff_pic_num_minus1 is present and corresponds to a negative difference to add to a picture number prediction value
1	abs_diff_pic_num_minus1 is present and corresponds to a positive difference to add to a picture number prediction value
2	long_term_pic_idx is present and specifies the long-term index for a reference picture
3	End loop for re-mapping of reference picture set relative indexing default order

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

abs_diff_pic_num_minus1 plus 1 indicates the absolute difference between the picture number of the picture being remapped and the picture number prediction value. When field_pic_flag is equal to 0, the value of abs_diff_pic_num_minus1 shall be in the range of 0 to MaxFrameNum -1, inclusive. When field_pic_flag is equal to 1, the value of abs_diff_pic_num_minus1 shall be in the range of 0 to 2*MaxFrameNum -1, inclusive. [Ed.Note: I'm not completely sure about this.]

long_term_pic_idx indicates the long-term picture index of the picture being remapped. In the case of frame -structured pictures, it shall be less than max_long_term_pic_idx_plus1; while decoding a field, it shall be less than 2 * max_long_term_pic_idx_plus1.

7.4.3.2 Prediction weight table semantics

[Ed.Note: the coupling of this with picture re -mapping is not yet worked out.]

luma_log_weight_denom is the binary logarithm of the denominator for all luma weighting factors. The value of luma_log_weight_denom shall be in t he range of 0 to 7, inclusive.

chroma_log_weight_denom is the binary logarithm of the denominator for all chroma weighting factors. The value of chroma_log_weight_denom shall be in the range of 0 to 7, inclusive.

luma_weight_l0_flag equal to 1 indicates that weighting factors are present for the luma component of list 0 prediction, while equal to 0 indicates that these weighting factors are not present. If luma_weight_l0_flag is equal to 0, then luma_weight_l0[i] shall be inferred as equal to $2^{luma\_log\_weight\_denom}$ and luma_offset_l0[i] shall be inferred as equal to 0.

luma_weight_l0[i] is the weighting factor applied to the luma prediction value for reference index i in list 0 prediction. Allowed range is –128 to 128, inclusive.

luma_offset_l0[i] is the additive offset applied to the luma prediction value for reference index i in list 0 prediction. Allowed range is –256 to 255, inclusive.

chroma_weight_l0_flag equal to 1 indicates that weighting factors are present for the Cb and Cr components of list 0 prediction while equal to 0 indicates that these weighting factors are not present. If chroma_weight_l0_flag is equal to 0 , chroma_weight_l0[i] shall be inferred as equal to $2^{chroma\_log\_weight\_denom}$ and chroma_offset_l0[i] shall be inferred as equal to 0.

chroma_weight_l0[i][0] is the weighting factor applied to the Cb prediction values for reference index i in list 0 prediction. Allowed range is –128 to 128, inclusive.

chroma_offset_l0[i][0] is the additive offset applied to the Cb pre diction values for reference index i in list 0 prediction. Allowed range is –256 to 255, inclusive.

chroma_weight_l0[i][1] is the weighting factor applied to the Cr prediction values for reference index i in list 0 prediction. Allowed range is –128 to 128, inclusive.

chroma_offset_l0[i][1] is the additive offset applied to the Cr prediction values for reference index i in list 0 prediction. Allowed range is –256 to 255, inclusive.

luma_weight_l1_flag equal to 1 indicates that weighting factors are present for the luma component of the list 1 prediction while equal to 0 indicates that these weighting factors are not present. If luma_weight_l1_flag is equal to 0, luma_weight_l1[i] shall be inferred as equal to $2^{luma\_log\_weight\_denom}$ and luma_offset_l1[i] shall be inferred as equal to 0.

luma_weight_l1[i] is the weighting factor applied to the luma prediction values for reference index i in list 1 prediction. Allowed range is –128 to 128, inclusive.

luma_offset_l1[i] is the additive offset applied to the luma prediction value for reference index i in list 1 prediction. Allowed range is –128 to 128, inclusive.

chroma_weight_l1_flag equal to 1 indicates that weighting factors are present for the chroma component of the list 1 prediction while equal to 0 indicates that these weighting factors are not present. If chroma_weight_l1_flag is equal to 0, chroma_weight_l1[i][j] shall be inferred as equal to $2^{chroma\_log\_weight\_denom}$ and chroma_offset_l1[i][j] shall be inferred as equal to 0.

chroma_weight_l1[i][0] is the weighting factor applied to the Cb prediction values for reference index i in list 1 prediction. Allowed range is –128 to 128, inclusive.

chroma_offset_l1[i][0] is the additive offset applied to the Cb prediction values fo r reference index i in list 1 prediction. Allowed range is –256 to 255, inclusive.

chroma_weight_l1[i][1] is the weighting factor applied to the Cr prediction values for reference index i in list 1 prediction. Allowed range is –128 to 128, inclusive.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

chroma_offset_l[i][1] is the additive offset applied to the Cr prediction values for reference index i in list 1 prediction. Allowed range is –256 to 255, inclusive.

7.4.3.3 Decoded picture buffer management semantics

The syntax elements num_ref_frames, adapt ive_ref_pic_buffering_mode_flag, memory_management_control_operation, difference_of_pic_nums_minus1, long_term_pic_idx, and max_long_term_pic_idx_plus1 specify the management of pictures in the decoded picture buffer. Further, the decoded picture buffer ca n be modified by marking reference pictures as "unused for reference", by assigning long -term picture indices and by removing [Ed.Note: Meaning?] reference and/or non-reference pictures from the decoded picture buffer. The syntax elements adaptive_ref_pic_ buffering_mode_flag, memory_management_control_operation, difference_of_pic_nums_minus1, long_term_pic_idx, and max_long_term_pic_idx_plus1 shall be identical for all coded slices of a coded picture.

adaptive_ref_pic_buffering_mode_flag specifies the reference picture buffering mode of the currently decoded picture and specifies how the decoded picture buffer is modified after the current picture is decoded. The values for adaptive_ref_pic_buffering_mode_flag are specified in Table 7-6.

Table 7-6 – Interpretation of adaptive_ref_pic_buffering_mode_flag

adaptive_ref_pic_buffering_mode_flag	Reference picture buffering mode specified
0	Sliding window reference picture buffering mode; A simple buffering mode providing a first-in first-out mechanism for reference pictures that are not assigned a long-term index
1	Adaptive reference picture buffering mode; A more flexible reference picture buffering mode than sliding window reference picture buffering mode providing syntax elements to specify marking reference pictures as "unused for reference", to assign long-term picture indices, and to remove pictures from the decoded picture buffer.

memory_management_control_operation specifies a control operation to be applied to manage the decoded picture buffer. The memory_management_control_operation parameter is followed by data necessary for the operation specified by the value of memory_management_control_operation. memory_management_control_oper ation commands do not affect the buffer contents or the decoding process for the decoding of the current frame. They specify the necessary buffer status for the decoding of subsequent coded pictures. The values and control operations associated with memory_management_control_operation are specified in Table 7-7.

If memory_management_control_operation is 5 (reset all reference pictures), all frames and fields marked as "used for prediction" in the decoded picture buffer shall be rem oved from the decoded picture buffer. Moreover, the maximum long-term reference picture index shall be reset to 0.

If memory_management_control_operation is 6 (reset all non -reference pictures), all frames and fields marked as "unused for prediction" shall be removed from the decoded picture buffer.

If memory_management_control_operation is 7 (reset all reference and non -reference pictures), all frames and fields shall be removed from the decoded picture buffer. Moreover, the maximum long -term picture index shall be reset to zero.

The frame height and width shall not change within [Ed.Note: meaning?] the bitstream except within a picture containing memory_management_control_operation commands 5 (reset all reference pictures) or 7 (reset all reference and no n-reference pictures).

If nal_ref_idc of the current picture is equal to 0, the value of the memory_management_control_operation shall not contain any of the following types of memory_management_control_operation commands:

a) Any reset all reference pictures, reset all non-reference pictures, and reset all reference and non -reference pictures memory_management_control_operation commands,

b) Any memory_management_control_operation command which marks any other reference picture (other than the current pictur e) as a non-reference pictures that has not also been previously marked as a non - reference pictures, or

c) Any memory_management_control_operation command which assigns a long -term index to a picture that has not also been assigned the same long -term index previously.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 7-7 – Memory management control operation (memory_management_control_operation) values

memory_management_control_operation	Memory Management Control Operation
0	End memory_management_control_operation Loop
1	Mark a short-term picture as "Unused for reference"
2	Mark a long-term picture as "Unused for reference"
3	Assign a long-term index to a picture
4	Specify the maximum long-term picture index
5	Reset all reference pictures
6	Reset all non-reference pictures
7	Reset all reference and non-reference pictures

difference_of_pic_nums_minus1 is used to assign a long-term index to a short-term reference picture or to mark a short-term reference picture as "unused for reference". When field_pic_flag is equal to 0, the value of difference_of_pic_nums_minus1 shall be in the range of 0 to MaxFrameNum -1, inclusive. When field_pic_flag is equal to 1, the value of difference_of_pic_nums_minus1 shall be in the range of 0 to 2*MaxFrameNum -1, inclusive. [Ed.Note: I'm not completely sure about this.]

long_term_pic_idx is used to assign a long-term index to a picture or to mark a long-term picture as "unused for reference". The value of long_term_pic_idx shall be in the range of 0 to max_long_term_pic_idx_plus1, inclusive.

max_long_term_pic_idx_plus1 minus 1 indicates the maximum index allowed for long-term reference frames (until receipt of another value of max_long_term_pic_idx_plus1). The decoder shall initially infer that max_long_term_pic_idx_plus1 is 0 until some other value has been received. The range of max_long_term_pic_idx_plus1 is specified in Annex A. [Ed.Note: specify in Annex A. If not specified otherwise, should be 0 to 16.]

7.4.4 Slice data semantics

cabac_alignment_bit is bit equal to 0. [Ed. Note: Shouldn't it be equal to 1? Seems foolish, as all other similar junk bits are set equal to 1, which is a better value for start-code emulation prevention.]

mb_skip_run is present only when entropy_coding_mode is equal to 0. mb_skip_run indicates the number of consecutive skipped macroblocks, in P slices, or number of consecutive macroblocks of Direct_16x16 macroblock type for which no additional transform coefficients are present, in B slices. The value of mb_skip_run is in the range of 0 to MaxMbAddress+1.

mb_skip_flag is present only when entropy_coding_mode is equal to 1. mb_skip_flag equal to 1 indicates that the next macroblock is skipped, in a P slice, or is of Direct_16x16 macroblock type without additional transform coefficients coded, in a B slice. mb_skip_flag equal to 0 indicates that the next macroblock is not skipped, in a P slice, or is not of Direct_16x16 macroblock type without additional transform coefficients coded, in a B slice.

mb_field_decoding_flag equal to 0 indicates that the current macroblock pair is decoded in frame decoding mode and equal to 1 indicates that the macroblock pair is decoded in field decoding mode.

end_of_slice_flag equal to 0 indicates that another macroblock is following, whereas end_of_slice_flag equal to 1 indicates the end of the slice and that no further macroblock follows. end_of_slice_flag is only present if entropy_coding_mode is equal to 1 for each macroblock except for the last macroblock of the picture.

When decoding a frame and mb_adaptive_frame_field_flag is equal to 1, the number of macroblocks, including skipped macroblocks, in the slice shall be an integer multiple of 2.

7.4.5 Macroblock layer semantics

mb_type indicates the macroblock type. The semantics of mb_type depend on the slice type.

Tables and semantics are specified for the various macroblock types for I, SI, P, SP, and B slices. Each table presents the value of mb_type, the name of mb_type, the number of macroblock partitions used (given by the num_mb_part(mb_type) function), the prediction mode of the macroblock (if it is not partitioned) or the first partition (given by the mb_part_pred_mode(mb_type, 1) function) and the prediction mode of the second partition (given by the

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

mb_part_pred_mode(mb_type, 2) function). When a value is not applicable it is designated by "na". A column of the table including only not applicable ("na") values does not appear in the table. In the text, the value of mb_type may be referred to as the macroblock type and a value X of mb_part_pred_mode may be referred to in the text by "X macroblock (partition) prediction mode" and the macroblock types may be referred to as "X prediction macroblocks".

Table 7-8 shows the allowed collective macroblock types for each slice_type. [E d.Note: It is proposed to delete the table as e.g. there are some macroblock types with Pred_L0 prediction mode that can not belong to a B slice]

Table 7-8 – Allowed collective macroblock types for slice_type

slice_type	allowed collective macroblock types
I (slice)	I (see Table 7-9) (macroblock types)
P (slice)	P (see Table 7-11) and I (see Table 7-9) (macroblock types)
B (slice)	B (see Table 7-12) and I (see Table 7-9) (macroblock types)
SI (slice)	SI (see Table 7-10) and I (see Table 7-9) (macroblock types)
SP (slice)	P (see Table 7-11) and I (see Table 7-9) (macroblock types)

Macroblock types which may be collectively referred to as I macroblock types are specified in Table 7-9.

The macroblock types for I slices are only I macroblock types.

Table 7-9 – Macroblock types for I slices

mb_type	Name of mb_type	mb_part_pred_mode (mb_type, 0)	Intra16x16 PredMode	CodedBlock PatternChroma	CodedAC PatternLuma
0	I_4x4	Intra_4x4	na	na	na
1	I_16x16_0_0_0	Intra_16x16	0	0	0
2	I_16x16_1_0_0	Intra_16x16	1	0	0
3	I_16x16_2_0_0	Intra_16x16	2	0	0
4	I_16x16_3_0_0	Intra_16x16	3	0	0
5	I_16x16_0_1_0	Intra_16x16	0	1	0
6	I_16x16_1_1_0	Intra_16x16	1	1	0
7	I_16x16_2_1_0	Intra_16x16	2	1	0
8	I_16x16_3_1_0	Intra_16x16	3	1	0
9	I_16x16_0_2_0	Intra_16x16	0	2	0
10	I_16x16_1_2_0	Intra_16x16	1	2	0
11	I_16x16_2_2_0	Intra_16x16	2	2	0
12	I_16x16_3_2_0	Intra_16x16	3	2	0
13	I_16x16_0_0_1	Intra_16x16	0	0	1
14	I_16x16_1_0_1	Intra_16x16	1	0	1
15	I_16x16_2_0_1	Intra_16x16	2	0	1
16	I_16x16_3_0_1	Intra_16x16	3	0	1
17	I_16x16_0_1_1	Intra_16x16	0	1	1
18	I_16x16_1_1_1	Intra_16x16	1	1	1
19	I_16x16_2_1_1	Intra_16x16	2	1	1

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

20	I_16x16_3_1_1	Intra_16x16	3	1	1
21	I_16x16_0_2_1	Intra_16x16	0	2	1
22	I_16x16_1_2_1	Intra_16x16	1	2	1
23	I_16x16_2_2_1	Intra_16x16	2	2	1
24	I_16x16_3_2_1	Intra_16x16	3	2	1

The following semantics are assigned to the macroblock types in Table 7-9:

I_4x4: the macroblock is coded as Intra_4x4 prediction macroblock.

I_16x16_0_0_0, I_16x16_1_0_0, I_16x16_2_0_0, I_16x16_3_0_0, I_16x16_0_1_0, I_16x16_1_1_0, I_16x16_2_1_0, I_16x16_3_1_0, I_16x16_0_2_0, I_16x16_1_2_0, I_16x16_2_2_0, I_16x16_3_2_0, I_16x16_0_0_1, I_16x16_1_0_1, I_16x16_2_0_1, I_16x16_3_0_1, I_16x16_0_1_1, I_16x16_1_1_1, I_16x16_2_1_1, I_16x16_3_1_1, I_16x16_0_2_1, I_16x16_1_2_1, I_16x16_2_2_1, I_16x16_3_2_1: the macroblocks is coded as Intra_16x16 prediction macroblock.

To each Intra_16x16 prediction macroblock, an Intra16x1 6PredMode is assigned which indicates the Intra_16x16 prediction mode, CodedBlockPatternChroma contains the coded block pattern bits for chroma as specified below in this subclause, and CodedACPatternLuma indicates whether non -zero AC levels are present. CodedACPatternLuma equal to 0 indicates that there are no AC transform levels in the luma component of the macroblock. CodedACPatternLuma equal to 1 indicates that at least one AC transform level is in the luma component of the macroblock, requiring scann ing of AC coefficient values for all 16 of the 4x4 blocks in the 16x16 block.

Intra_4x4 specifies the macroblock prediction mode and indicates that the Intra_4x4 prediction process is invoked as specified in subclause 8.3.18.3.2. Intra_4x4 is an Intra macroblock prediction mode.

Intra_16x16 specifies the macroblock prediction mode and indicates that the Intra_16x16 prediction process is invoked as specified in subclause 8.3.2. Intra_16x16 is an Intra macroblock prediction mode.

A macroblock type which may be referred to as SI macroblock type is specified in Table 7-10.

The macroblock types for SI slices are specified in Table 7-10 and Table 7-9. The mb_type value 0 is specified in Table 7-10 and the mb_type values 1 to 25 are specified in Table 7-9, indexed by subtracting 1 from the value of mb_type.

Table 7-10 – Macroblock type with value 0 for SI slices

mb_type	Name of mb_type	mb_part_pred_mode(mb_type, 0)
0	SI	Intra_4x4

The following semantics are assigned to the macroblock type in Table 7-10: the SI macroblock is coded as Intra_4x4 prediction macroblock.

Macroblock types which may be collectively referred to as P macroblock types are specified in Table 7-11.

The macroblock types for P and SP slices a re specified in Table 7-11 and Table 7-9. The mb_type values 0 to 4 are specified in Table 7-11 and the mb_type values 5 to 29 are specified in Table 7-9, indexed by subtracting 5 from the value of mb_type.

Table 7-11 – Macroblock type values 0 to 4 for P and SP slices

mb_type	Name of mb_type	num_mb_part (mb_type)	mb_part_pred_mode (mb_type, 0)	mb_part_pred_mode (mb_type, 0)
0	P_L0_16x16	1	Pred_L0	na
1	P_L0_L0_16x8	2	Pred_L0	Pred_L0
2	P_L0_L0_8x16	2	Pred_L0	Pred_L0
3	P_8x8	4	na	na

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

4	P_8x8ref0	4	na	na

The following semantics are assigned to the macroblock types in Table 7-11:

P_L0_16x16, P_L0_L0_16x8, P_L0_L0_8x16, and P_8x8: the macroblock is predicted from a past picture with luma block sizes 16x16, 16x8, 8x16, and 8x8, respectively, and the associated chroma blocks. For the macroblock types NxM=16x16, 16x8, and 8x16, a moti on vector difference is decoded for each NxM luma block and the associated chroma blocks. If N=M=8, for each sub-macroblock an additional syntax element is decoded which indicates in which type the corresponding sub-macroblock is decoded (see subclause 7.4.5.2). Depending on N, M and the sub-macroblock types there may be 1 to 16 sets of motion vector difference for the for a macroblock.

P_8x8ref0: same as P_8x8 but ref_idx_l0 is not present in the bitstream and set equal to 0 f or all sub-macroblocks. P_8x8ref0 is only present when entropy_coding_mode is equal to 0.

Pred_L0 specifies the macroblock (partition) prediction mode and indicates that the Inter prediction process is invoked using list 0 prediction. Pred_L0 is an Inter m acroblock prediction mode.

Macroblock types which may be collectively referred to as B macroblock types are specified in Table 7-12.

The macroblock types for B slices are specified in Table 7-12 and Table 7-9. The mb_type values 0 to 22 are specified in Table 7-12 and the mb_type values 23 to 47 are specified in Table 7-9, indexed by subtracting 23 from the value of mb_type.

Table 7-12 – Macroblock type values 0 to 22 for B slices

mb_type	Name of mb_type	num_mb_part (mb_type)	mb_part_pred_mode (mb_type, 0)	mb_part_pred_mode (mb_type, 1)
0	B_Direct_16x16	1	Direct	na
1	B_L0_16x16	1	Pred_L0	na
2	B_L1_16x16	1	Pred_L1	na
3	B_Bi_16x16	1	BiPred	na
4	B_L0_L0_16x8	2	Pred_L0	Pred_L0
5	B_L0_L0_8x16	2	Pred_L0	Pred_L0
6	B_L1_L1_16x8	2	Pred_L1	Pred_L1
7	B_L1_L1_8x16	2	Pred_L1	Pred_L1
8	B_L0_L1_16x8	2	Pred_L0	Pred_L1
9	B_L0_L1_8x16	2	Pred_L0	Pred_L1
10	B_L1_L0_16x8	2	Pred_L1	Pred_L0
11	B_L1_L0_8x16	2	Pred_L1	Pred_L0
12	B_L0_Bi_16x8	2	Pred_L0	BiPred
13	B_L0_Bi_8x16	2	Pred_L0	BiPred
14	B_L1_Bi_16x8	2	Pred_L1	BiPred
15	B_L1_Bi_8x16	2	Pred_L1	BiPred
16	B_Bi_L0_16x8	2	BiPred	Pred_L0
17	B_Bi_L0_8x16	2	BiPred	Pred_L0
18	B_Bi_L1_16x8	2	BiPred	Pred_L1
19	B_Bi_L1_8x16	2	BiPred	Pred_L1
20	B_Bi_Bi_16x8	2	BiPred	BiPred

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

21	B_Bi_Bi_8x16	2	BiPred	BiPred
22	B_8x8	4	na	na

The macroblock types in Table 7-12 are collectively referred to a B macroblock types.

The following semantics are assigned to the macroblock types in Table 7-12:

B_Direct_16x16 type: no motion vector data or reference picture index is transmitted.

B_x_y_NxM, x,y=L0,L1,Bi: each N xM luma block and the associated chroma blocks of a macroblock are predicted by using decoded motion vector differences and reference pictures. As shown in Table 7-12, for 16x16 macroblock partitions, 3 different macroblock types that differ in their prediction modes can be decoded for a macroblock. For the 16x8 and 8x16 macroblock partitions, 18 different combinations of prediction modes can be decoded for a macroblock. If a macroblock is coded as B_8x8 macroblock type, an addition al codeword for each 8x8 macroblock partition, referred to as sub-macroblock, indicates the decomposition of the sub -macroblock (see Table 7-19).

B_8x8: the macroblock is partitioned into sub -macroblocks. The coding [Ed.Note: Meani ng?] of eacb sub-macroblock is specified using sub_mb_type.

Direct specifies the macroblock (partition) prediction mode and indicates that no data are present in the bitstream for the prediction process. The variables for the Inter prediction process are d erived may be using list 0 or list 1 prediction. Direct is an Inter macroblock prediction mode.

Pred_L1 specifies the macroblock (partition) prediction mode and indicates that the Inter prediction process is invoked using list 1 prediction. Pred_L1 is an I nter macroblock prediction mode.

BiPred specifies the macroblock (partition) prediction mode and indicates that the Inter prediction process is invoked using list 0 and list 1 prediction. BiPred is an Inter macroblock prediction mode.

pcm_alignment_bit is a bit equal to 1.

pcm_byte is a sample value. pcm_byte shall not be equal to 0. The first 256 pcm_byte values represent luma sample values in raster scan order within the macroblock. The next 64 pcm_byte values represent Cb sample values in raster scan order within the macroblock. The last 64 pcm_byte values represent Cr sample values in raster scan order within the macroblock.

coded_block_pattern specifies which of the 8x8 blocks - luma and chroma - contain transform coefficients. An indication that an 8x8 block contains coefficients means that one or more of its four 4x4 blocks contains coefficients'. The four least significant bits of coded_block_pattern contain information on which of four 8x8 luma blocks in the macroblock contains nonzero coeffici ents. These four bits are denoted as CodedBlockPatternLuma. The inverse scan of 8x8 blocks in the luma component of the macroblock is an inverse raster scan. A bit equal to zero in position n of coded_block_pattern (binary representation) indicates that the corresponding 8x8 block has no coefficients and a bit equal to 1 indicates that the 8x8 block has one or more non -zero coefficients.

The two most significant bits of coded_block_pattern contain information on the chroma blocks. The 3 possibilities ar e specified in Table 7-13.

Table 7-13 – Specification of CodedBlockPatternChroma values

CodedBlockPatternChroma	Description
0	All chroma coefficients are 0.
1	One or more chroma DC coefficients are non-zero. All chroma AC coefficients are 0.
2	Zero or more chroma DC coefficients are non-zero. One or more chroma AC coefficients are non-zero.

The value of coded_block_pattern for a macroblock is given by

coded_block_pattern = CodedBlo ckPatternLuma + 16 × CodedBlockPatternChroma

[Ed.Note: need for equations for CodedBlockPatternLuma and CodedBlockPatternChroma as there is a "missing link" between bits and unsigned integer value through unsigned integer binary representation]

Table 7-14 shows the coded_block_pattern for macroblock prediction modes Intra_4x4 and Inter.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 7-14 – Assignment of coded_block_pattern codewords for macroblock prediction types

coded_block_pattern	coded_block_pattern assignment to macroblock prediction modes	
	Intra_4x4	Inter
0	47	0
1	31	16
2	15	1
3	0	2
4	23	4
5	27	8
6	29	32
7	30	3
8	7	5
9	11	10
10	13	12
11	14	15
12	39	47
13	43	7
14	45	11
15	46	13
16	16	14
17	3	6
18	5	9
19	10	31
20	12	35
21	19	37
22	21	42
23	26	44
24	28	33
25	35	34
26	37	36
27	42	40
28	44	39
29	1	43
30	2	45
31	4	46

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

32	8	17
33	17	18
34	18	20
35	20	24
36	24	19
37	6	21
38	9	26
39	22	28
40	25	23
41	32	27
42	33	29
43	34	30
44	36	22
45	40	25
46	38	38
47	41	41

mb_qp_delta the value of QP_Y can be changed in the macroblock layer by the parameter mb_qp_delta. The mb_qp_delta parameter is present only for non -skipped macroblocks, as specified by:

- If coded_block_pattern indicates that there are nonzero transform coefficients in the macroblock or
- If the macroblock is 16x16 based intra coded

The decoded value of mb_qp_delta shall be in the range from -26 to +25, inclusive. This specifies the value of QP_Y in the range [0..51], as given by

$$QP_Y = (QP_Y + mb\_qp\_delta + 52) \% 52 \quad (7\text{-}12)$$

7.4.5.1 Macroblock prediction semantics

All samples of the macroblock are predicted. The prediction modes are derived using the following syntax elements.

prev_intra4x4_pred_mode_flag[4x4LumaBlkIdx] and **rem_intra4x4_pred_mode**[4x4LumaBlkIdx] indicate the Intra_4x4 prediction of the 4x4 luma block with index 4x4LumaBlkIdx.

intra_chroma_pred_mode indicates the type of spatial prediction used for chroma whenever any part of t he luma macroblock is intra coded. This is shown in Table 7-15.

Table 7-15 – Relationship between intra_chroma_pred_mode and spatial prediction modes

intra_chroma_pred_mode	[Ed.Note Intra_16x16] Prediction Mode
0	DC
1	Horizontal
2	Vertical
3	Plane

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

ref_idx_l0, when present, indicates the reference picture to be used for prediction. The value of ref_idx_l0 shall be in the range of 0 to num_ref_idx_l0_active_minus1, inclusive. [Ed.N ote: relationship with weighted prediction may not be fully worked out yet.]

If field_pic_flag indicates that the current picture is a frame picture, then the reference picture is a previous frame in list 0 that was either indicated as a single frame pictu re or a frame that was indicated as two field pictures and has been reconstructed as a frame. [Ed.Note: Indexing into an ordered list is simple and does not need the following tables.] For reference frame indexing, Table 7-16 illustrates the reference frame:

Table 7-16 – Reference frame indicated by ref_idx_10 for frame picture (informative)

ref_idx_10	Reference frame
0	The first frame in list 0
1	The second frame in list 0
2	The third frame in list 0
...	...

If field_pic_flag indicates that the current picture is a field picture, then the reference picture is a previous field in list 0 that was either coded as part of a frame-structured picture or coded as a field-structured picture. [Ed. Note: What about field macroblocks in a frame picture in MB-AFF?] For reference field indexing, Table 7-17 illustrates the reference field:

Table 7-17 – Reference field indicated by ref_idx_10 for field picture (informative)

ref_idx_10	Reference field
0	The first field in list 0
1	The second field in list 0
2	The third field in list 0
...	...

If num_ref_idx_l0_active_minus1 is equal to 0, ref_idx_l0 is not present. If num_ref_i dx_l0_active_minus1 is equal to 1, only a single encoded bit is used to represent ref_idx_l0. If num_ref_idx_l0_active_minus1 is greater than 1, the value of ref_idx_l0 is represented by a decoded index.

ref_idx_l1 has the same semantics as ref_idx_l0, ex cept that it is applied to the reference index list 1. The value of ref_idx_l1 shall be in the range of 0 to num_ref_idx_l1_active_minus1, inclusive. [Ed.Note: relationship with weighted prediction may not be fully worked out yet.]

mvd_l0 indicates the difference between a vector component to be used and its prediction. The horizontal motion vector component difference is decoded first in decoding order. If so indicated by mb_type, vector data for 1 to 16 blocks are transmitted. For the coded motion vector, a prediction is formed for the horizontal and vertical components of the motion vector. The prediction for the motion vector component is added to the motion vector component difference mvd_l0 to form the motion vector component used in inter prediction. The range of mvd_l0 is constrained by constraints on decoded motion vector values specified in Annex A.

mvd_l1 has the same semantics as mvd_l0, except that it is applied to the reference index list 1. The range of mvd_l1 is constrained by constraints on motion vector values specified in Annex A.

7.4.5.2 Sub-macroblock prediction semantics

sub_mb_type indicates the sub-macroblock types.

Tables and semantics are specified for the various sub -macroblock types for I, SI, P, SP, and B slices. Each table presents the value of sub_mb_type, the name of sub_mb_type, the number of sub -macroblock partitions used (given by the num_sub_mb_part(sub_mb_type) function) and the prediction mode of the sub-macroblock (given by the sub_mb_part_pred_mode(sub_mb_type) function). In the text, the value of sub_mb_type may be referred to in the text by "sub-macroblock type". In the text, the value of sub_mb_part_pred mode may be referred to by "[Ed.Note: sub -macroblock] prediction mode".

The sub-macroblock types for P macroblock type s are specified in Table 7-18.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 7-18 – Sub-macroblock types in P macroblocks

sub_mb_type	Name of sub_mb_type	num_sub_mb_part (sub_mb_type)	sub_mb_pred_mode (sub_mb_type)
0	P_L0_8x8	1	Pred_L0
1	P_L0_8x4	2	Pred_L0
2	P_L0_4x8	2	Pred_L0
3	P_L0_4x4	4	Pred_L0

The following semantics are assigned to the sub -macroblock types in Table 7-18:

P_L0_ XxY, X,Y=4,8 the corresponding partition of the sub -macroblock is predicted from a past picture with luma block size 8x8, 8x4, 4x8, and 4x4, respectively, and the associated chroma blocks. A motion vector is transmitted for each NxM=8x8, 8x4, 4x8, and 4x4 block. Depending on N and M, up to 4 motion vector component differences may be decoded for a sub-macroblock, and thus up to 16 motion vector component differences maybe dedoced for a macroblock.

The sub-macroblock types for B macroblock types are specified in Table 7-19.

Table 7-19 – Sub-macroblock types in B macroblocks

sub_mb_type	Name of sub_mb_type	num_sub_mb_part (sub_mb_type)	sub_mb_pred_mode (sub_mb_type)
0	B_Direct_8x8	1	Direct
1	B_L0_8x8	1	Pred_L0
2	B_L1_8x8	1	Pred_L1
3	B_Bi_8x8	1	BiPred
4	B_L0_8x4	2	Pred_L0
5	B_L0_4x8	2	Pred_L0
6	B_L1_8x4	2	Pred_L1
7	B_L1_4x8	2	Pred_L1
8	B_Bi_8x4	2	BiPred
9	B_Bi_4x8	2	BiPred
10	B_L0_4x4	4	Pred_L0
11	B_L1_4x4	4	Pred_L1
12	B_Bi_4x4	4	BiPred

The following semantics are assigned to the sub-macroblock types in Table 7-19:

Pred_L0_XxY, X,Y=4,8, have the same semantics as in Table 7-18.

B_Z_X_Y, Z=L1,Bi, X,Y=4,8 [Ed.Note: I need input here !!!]

7.4.5.3 Residual data semantics

When entropy_coding_mo de is equal to 0, residual_block is equal to residual_block_cavlc which is used for parsing the syntax elements for transform levels.

When entropy_coding_mode is equal to 1, residual_block is equal to residual_block_cabac which is used for parsing the syntax elements for transform levels.

Depending on mb_type, luma or chroma, residual_block(Level, X) is invoked with the arguments Level and X. Level is an array containing the X transform levels that are parsed.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

When mb_part_pred_mode(mb_type, 0) is equal to Intra_16x16, the transform levels are parsed into two arrays: Intra16x16DCLevel and Intra16x16ACLevel. Intra16x16DCLevel contains the 16 transform levels of the DC transform levels for each 4x4 luma block. Intra16x16ACLevel[i8x8 * 4 + i4x4] contains for each 4x4 luma block indexed by i8x8 * 4 + i4x4 the 15 AC coefficients.

When mb_part_pred_mode(mb_type, 0) is equal not to Intra_16x16, the 16 transform levels for each 4x4 luma block indexed by i8x8 * 4 + i4x4 are contained in LumaResidualLevel[i8x8 * 4 + i4x4].

The 4 DC coefficients of all 4x4 chroma blocks per chroma component are parsed into ChromaDCLevel[CbCrIdx] with CbCrIdx indicating the chroma component.

The 15 AC coefficients for each 4x4 chroma coefficient which is indexed by i4x4 for ea ch chroma block and CbCrIdx for each chroma component are contained in ChromaACLevel[CbCrIdx][i4x4].

7.4.5.3.1 Residual 4x4 block CAVLC semantics

coeff_token indicates the total number of non-zero coefficients and the number of trailing one coefficients in a coefficient scan. A trailing one coefficient is one of up to three consecutive non -zero coefficients at the end of the zig-zag scan of non-zero coefficients that have a level with an absolute value of 1. The range of coeff_token is specified in subclause 9.2.

trailing_ones_sign[i **]** indicates the sign of the $(i+1)^{th}$ trailing one coefficient in a zig-zag scan. A trailing_ones_sign equal to 0 indicates that the level of the corresponding transform coefficient is +1. A trailing_ones_si gn equal to 1 indicates that the level of the corresponding transform coefficient is -1.

coeff_level[i **]** indicates the level and sign of the $(i+1)^{th}$ non-zero transform coefficient. The range of coeff_level is specified in subclause 9.2.

total_zeros indicates the total number of zero -valued coefficients that are located before the position of the last non -zero coefficient in a zig-zag or field scan of coefficients. The range of total_zeros is specified in subclause 9.2.

run_before[i **]** indicates the number of consecutive coefficient scanning positions with zero -valued coefficients before the $(i+1)^{th}$ non-zero coefficient in the zig-zag scan of coefficients. range of run_before[i] is specified in subcla use 9.2.

Level contains MaxNumCoeff transform levels for the current array.

7.4.5.3.2 Residual 4x4 block CABAC semantics

coded_block_flag indicates whether the block contains non -zero transform coefficients. If coded_block_flag is equal t o 0, the block contains no non -zero transform coefficients. If coded_block_flag is equal to 1, the block contains at least one non-zero transform coefficient.

significant_coeff_flag[i **]** indicates whether the transform coefficient at scanning position i is non-zero. If significant_coeff_flag[i] is equal to 0, the transform coefficient at scanning position i is equal to zero; if significant_coeff_flag[i] is equal to 1, the transform coefficient at scanning position i has a non -zero value.

last_significant_coeff_flag[i **]** indicates for the scanning position i whether there are non -zero transform coefficients for subsequent scanning positions i+1 to max_numcoeff -1. If all following transform coefficients (in scanning order) of the block have value equal to zero last_significant_coeff_flag[i] is equal to 1. If last_significant_coeff_flag[i] is equal to 0, there are further non-zero transform coefficients along the scanning path.

coeff_absolute_value_minus1 is the absolute value of the transform coefficien t minus 1. The value of coeff_absolute_value_minus1 shall be in the range of 0 to ???. [Ed.Note: 255*16 or smaller. Anything that obeys the constraints expressed in the inverse tranform decoding process to work is OK. No further contraint needs to be specified.]

coeff_sign_flag indicates the sign of the transform coefficient. If coeff_sign_flag is equal to 0, the corresponding transform coefficient has a positive value. If coeff_sign_flag is equal to 1, the corresponding transform coefficient has a negative value.

Level contains MaxNumCoeff transform levels for the current array.

8 Decoding process

Input to this process are syntax elements or variables from clause 7.

Output of this process are decoded samples.

This clause describes the decoding process given syntax elements or variables from clause 7.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

The decoding process is defined such that all decoders shall produce numerically identical results. Any decoding process that produces identical results to the process described here conforms to the decoding process requirements of this Recommendation | International Standard.

Subclause 8.1 describes the decoding process for NAL units with nal_unit_type equal to 1 through 10. [Ed.Note: for SEI and VUI only syntax and semantics, no decoding process]

Subclause 8.2 describes the decoding process for NAL units with nal_unit_type equal to 1 through 5.

Subclause 8.3 describes the decoding process for slice data in NAL units with nal_unit_type equal to 1, 2, and 5.

Subclause 8.4 describes the decoding process for slice data in NAL units with nal_unit_type equal to 1 and 2.

Subclause 8.5 describes the decoding process for slice data in NAL units with nal_unit_type equal to 1, 3, 4 and 5.

Subclause 8.6 describes the decoding process for slice data in NAL units with nal_unit_type equal to 1, 3, 4 and 5.

Subclause 8.7 describes the decoding process for NAL units with nal_unit_type equal to 1 through 5.

[Ed.Note: specify necessary order.]

8.1 NAL unit decoding process

Input to this process are NAL units.

Output of this process is ??? [Ed.Note: specify output of this process]

[Ed. Note: Waiting for input.]

8.2 Slice decoding process

8.2.1 Detection process of coded picture boundaries

Input to this process are the slice header of the slice to be decoded, nal_ref_idc of the NAL unit containing the slice to be decoded, the picture order count for the slice to be decoded, the slice header of the previously decoded slice, the nal_ref_idc of the NAL unit that contained the previously decoded slice, and the picture order count of the previously decoded slice.

Output of this process is an indication if decoding of a new picture is started from the slice to be decoded.

Decoding of a new picture is started from the slice to be decoded, herein called the current slice, if the slice is not a redundant slice and if one of the following conditions is true:

1. The frame number of the current slice is different from the frame number of the previously decoded slice.
2. The field_pic_flag of the previously decoded slice is different from the field_pic_flag of the current slice.
3. The bottom_field_flag exists in both the previously decoded slice and the current slice but is not equal in the two slices, or the bottom_field_flag exists in only one of the previously decoded slice and the current slice.
4. The nal_ref_idc of the previously decoded slice is different from the nal_ref_idc of the current slice, and either nal_ref_idc is equal to 0.
5. The frame number of the current slice is the same as the frame number in the previously decoded slice, and either delta_pic_order_cnt[0] is different from the delta_pic_order_cnt[0] in the previously decoded slice, or delta_pic_order_cnt[1] is different from the delta_pic_order_cnt[1] in the previously decoded slice
6. The frame number of the current slice is the same as the previously de coded slice, and pic_order_cnt_type is less than 2, and the calculated picture order counts are different for the two slices.

8.2.2 Decoding process for picture order count

Inputs to this process are pic_order_cnt_type, pic_order_present_flag, frame_num, field_pic_flag, bottom_field_flag, num_ref_frames_in_pic_order_cnt_cycle, delta_pic_order_cnt[], offset_for_non_ref_pic, offset_for_ref_frame[], and offset_for_top_to_bottom_field.

Outputs of this process are TopPicOrderCnt, BottomPicOrderCnt, FramePicOrderCnt, P icOrder, and DiffPicOrderCount.

Each field is associated with a picture order count, called TopPicOrderCnt for a top field and BottomPicOrderCnt for a bottom field, which indicates the output order of each field relative to the first output field of last 1 DR picture in decoding order. Each coded frame is associated with two picture order counts, called TopPicOrderCnt and BottomPicOrderCnt, which respectively indicate the output orders of its top field and bottom field relative to first output

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

field of the last IDR picture in decoding order. In addition, for a frame the Frame PicOrderCnt is defined to be the smaller of TopPicOrderCnt and BottomPicOrderCnt for the fields in the frame; if only one of TopPicOrderCnt and BottomPicOrderCnt is defined in the frame, then that value is the FramePicOrderCnt. TopPicOrderCnt and BottomPicOrderCnt are defined for every field in the decoded picture buffer.

Picture order counts are used to determine default index orderings for reference pictures in B slices (subclause 8.2.6.3.3 and 8.2.6.3.4), to represent picture order differences between frames and fields for vector scaling in motion vector prediction (subclause 8.4.1.2.2) and for implicit mode weighted prediction in B slices (subclause 0 [Ed. Note: Bad cross-reference]), and to determine when successive slices belong to different pictures (subclause 8.2.1).

The bitstream shall not contain data that results in a FramePicOrderCnt not equal to 0 for an IDR picture. Thus one of TopPicOrderCnt and BottomPicOrderCnt must be 0 for the fields of a frame whose FramePicOrderCnt is equal to zero. The FramePicOrderCnt, TopPicOrderCnt, and Bottom PicOrderCnt of each reference frame shall be stored as long as the frame stays in the decoded picture buffer.

The signed picture order difference from X to Y, DiffPicOrderCount(X, Y), between two fields or frames is defined as follows:

```
if (Z is a frame) PicOrder(Z) = FramePicOrderCnt of frame Z
else if (Z is a top field) PicOrder(Z) = TopPicOrderCnt of frame containing Z          (8-1)
else if (Z is a bottom field) PicOrder(Z) = BottomPicOrderCnt of frame containi ng Z
```

```
DiffPicOrderCount(X,Y) = PicOrder(X) - PicOrder(Y)                                    (8-2)
```

The bitstream shall not contain data that results in values of DiffPicOrderCount(X,Y) used in the decoding process that exceed the range from -2^{15} to $2^{15}-1$, inclusive.

If X is the current picture and Y and Z are reference pictures, Y and Z are considered to be in the same temporal direction from X if both DiffPicOrderCount(X,Y) and DiffPicOrderCount(X,Z) are positive or both are negative.

There are three modes for coding picture order counts. [Ed. Note: Modes or types? See the drafting guide. There seems to be a mix of these terms (e.g., see heading of next subclause).] The first two modes, mode 0 and mode 1, use values in the sequence parameter set to predict the picture order count for each field and encode only the difference between the predicted value and the true picture order count in the slice header. Mode 0 predicts the value of the picture order count from the picture order count of the most recently decoded reference picture. Mode 1 predicts the value of the picture order count from the picture order count of the most recent IDR picture. Thus both modes 0 and 1 are resilient to the loss of non-reference pictures. In addition, mode 1 is re silient to the loss of non-IDR reference pictures. In the third mode, mode 2, no picture order count information is present . Mode 2 shall not be used in a sequence that contains B slices and shall not be used in a sequence that contains two ore more cons ecutive non-reference frames in decoding order.

When field_pic_flag is equal to 0 and pic_order_present_flag is equal to 0, delta_pic_order_cnt[0] is coded but delta_pic_order_cnt[1] is not coded and is inferred to be equal to 0. If field_pic_flag is equal to 0 and pic_order_present_flag is equal to 1, then both delta_pic_order_cnt[0] and delta_pic_order_cnt[1] are coded; delta_pic_order_cnt[0] is used to calculate TopPicOrderCnt, and delta_pic_order_cnt[1] is used to calculate BottomPicOrderCnt.

The bitstream shall not contain data that results in values of Frame PicOrderCnt, TopPicOrderCnt, and BottomPicOrderCnt, FrameNumOffset, AbsFrameNum, or PicOrderCntCycleCnt used in the decoding process as specified in subclauses 8.2.2.1 and 8.2.2.2 that exceed the range of values from 0 to $2^{32}-1$, inclusive.

8.2.2.1 Decoding process for picture order count type 0

Inputs to this process are pic_order_cnt_type, pic_order_present_flag, field_pic_flag, bottom_field_flag, delta_pic_order_cnt[], offset_for_non_ref_pic, offset_for_ref_frame[].

Outputs of this process are TopPicOrderCnt, BottomPicOrderCnt, FramePicOrderCnt.

If pic_order_cnt_type is 0, the calculations in this subclause shall be used for TopPicOrderCnt and BottomPicOrderCnt.

If the pic_order_present_flag is equal to 0 or field_pic_flag is equal to 1 , delta_pic_order_cnt[1] is not coded and is set to zero in the following equations.

For an IDR picture, the values TopPreviousPicOrderCnt and BottomPrevio usPicOrderCnt are defined to be 0 in the equations below. Otherwise, TopPreviousPicOrderCnt is defined to be equal to the TopPicOrderCnt of the top field of the most recently decoded reference frame or field, and BottomPicOrderCnt is defined to be equal t o the BottomPicOrderCnt of the bottom field of the most recently decoded reference frame or field.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

If the picture is a non-reference picture, then OffsetPicOrderCnt is equal to offset_for_non_ref_pic, and if it is a reference picture then OffsetPicOrderCnt is equal to offset_for_ref_frame[0].

For a frame picture (i.e., when field_pic_flag is equal to 0), TopPicOrderCnt and BottomPicOrderCnt are calculated according to Equations 8-3 and 8-4.

$$\text{TopPicOrderCnt} = \text{TopPreviousPicOrderCnt} + \text{OffsetPicOrderCnt} + \text{delta\_pic\_order\_cnt}[\,0\,] \quad (8\text{-}3)$$

$$\text{BottomPicOrderCnt} = \text{TopPicOrderCnt} + \text{offset\_for\_top\_to\_bottom\_field} + \text{delta\_pic\_order\_cnt}[\,1\,] \quad (8\text{-}4)$$

For a top field picture (i.e., when field_pic_flag is equal to 1 and bottom_field_flag is equal to 0), TopPicOrderCnt is calculated according to Equation 8-3, and BottomPicOrderCnt is not changed (since, if present, it is coded in a different field picture).

For a bottom field picture (i.e., when field_pic_flag is equal to 1 and bottom_field_flag is equal to 1), BottomPicOrderCnt is calculated according to Equation 8-5, and TopPicOrderCnt is not changed (since, if present, it is coded in a different field picture).

$$\text{BottomPicOrderCnt} = \text{BottomPreviousPicOrderCnt} + \text{OffsetPicOrderCnt} + \text{delta\_pic\_order\_cnt}[\,0\,] \quad (8\text{-}5)$$

8.2.2.2 Decoding process for picture order count type 1

Input to this process are pic_order_cnt_type, pic_order_present_flag, frame_num, field_pic_flag, bottom_field_flag, num_ref_frames_in_pic_order_cnt_cycle, delta_pic_order_cnt[], offset_for _non_ref_pic, offset_for_ref_frame[], offset_for_top_to_bottom_field.

Output of this process are TopPicOrderCnt, BottomPicOrderCnt, FramePicOrderCnt.

If pic_order_cnt_type is 1, the calculations in this subclause shall be used for TopPicOrderCnt and Bottom PicOrderCnt.

TopPicOrderCnt and BottomPicOrderCnt are calculated as follows, wherePreviousFrameNum is the frame_num of the previous picture in decoding order.

First, the FrameNumOffset is updated according to

```
if ( picture is an IDR picture )
    FrameNumOffset = 0
else if (frame_num < PreviousFrameNum)                    (8-6)
    FrameNumOffset = FrameNumOffset + MaxFrameNum
```

Second, the AbsFrameNum (the frame number relative to the last IDR picture) is calculated according to:

```
if( num_ref_frames_in_pic_order_cnt_cycle != 0 )
  AbsFrameNum = FrameNumOffset + frame_num
else                                                      (8-7)
  AbsFrameNum = 0
if( nal_ref_idc == 0 && AbsFrameNum > 0)
  AbsFrameNum = AbsFrameNum – 1
```

Third, if AbsFrameNum > 0, the PicOrderCntCycleCnt and FrameNumInPicOrderCntCycle are calculated according to:

```
if(AbsFrameNum > 0) {
  PicOrderCntCycleCnt =
       (AbsFrameNum – 1) / num_ref_frames_in_pic_order_cnt_cycle   (8-8)
  FrameNumInPicOrderCntCycle =
      (AbsFrameNum – 1) % num_ref_frames_in_pic_order_cnt_cycle
}
```

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

If the pic_order_present_flag is equal to 0 or field_pic_flag is equal to 1, delta_pic_order_cnt[1] is not coded and is inferred to be equal to 0 in the following equations.

In the following, the ExpectedDeltaPerPicOrderCntCycle is the sum of offset_for_ref_frame values (num_ref_frames_in_pic_order_cnt_cycle of these were coded in the sequence header).

Finally, the TopPicOrderCnt and /or BottomPicOrderCnt of th e picture are computed using the algorithm expressed as:

```
if( AbsFrameNum > 0 ){
        ExpectedPicOrderCnt =
            PicOrderCntCycleCnt * ExpectedDeltaPerPicOrderCntCycle
        for( i = 0; i <= FrameNumInPicOrderCntCycle; i++ )
            ExpectedPicOrderCnt = ExpectedPicOrderCnt + offset_for_ref_frame[ i ]
} else
        ExpectedPicOrderCnt = 0
if( nal_ref_idc == 0 )                                                    (8-9)
        ExpectedPicOrderCnt = ExpectedPicOrderCnt + offset_for_non_ref_pic
if( field_pic == 0 ) {
        TopPicOrderCnt = ExpectedPicOrderCnt + delta_pic_order_cnt[ 0 ]
        BottomPicOrderCnt = TopPicOrderCnt + offset_for_top_to_bottom_field
                    + delta_pic_order_cnt[ 1 ]
}else if ( bottom_field_flag == 0 )
        TopPicOrderCnt = ExpectedPicOrderCnt + delta_pic_order_cnt[ 0 ]
else if ( bottom_field_flag == 1 )
        BottomPicOrderCnt = ExpectedPicOrderCnt + offset_for_top_to_bottom_field
                    + delta_pic_order_cnt[ 0 ]
```

8.2.2.3 Examples for the decoding process for picture order count types 0 and 1 (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Type 0 can code the picture order counts quite efficiently when the encoder is reordering the pictures from display order to coding order in a regular pattern.

For a first example, assume there is a constant frame rate and frames are coded in display order as IPPPPPP... where the I frame is an IDR picture and all the P frames are reference pictures. Also assume that frames are not interlaced, so that TopPicOrderCnt and BottomPicOrderC nt are equal for each frame. In this case, mode 0 can be set up with offset_for_ref_frame[0] = 1 and and pic_order_present_flag = 0. In this case, the frame _num will increase by 1 with each picture, and OffsetPicOrderCnt will always correctly predict t he picture order count, so that delta_pic_order_cnt[0] will be zero for each frame. If delta_pic_order_always_zero_flag is set to 1, then no bits need be sent for delta_pic_order_cnt[0] or delta_pic_order_cnt[1]. The Table 8-1 illustrates how this works for the first few frames.

Table 8-1 – First example for picture order count type 0 [Ed. Note: Editorial issue with first column – Do we have frame types? What purpose does that column if any?]

Frame Type	frame_num	FramePicOrderCnt	TopPicOrderCnt	BottomPicOrderCnt	delta_pic_order_cnt[0]	delta_pic_order_cnt[1]
I	0	0	0	0	0 not sent	0 not sent
P	1	1	1	1	0 not sent	0 not sent
P	2	2	2	2	0 not sent	0 not sent
P	3	3	3	3	0 not sent	0 not sent
P	4	4	4	4	0 not sent	0 not sent
P	5	5	5	5	0 not sent	0 not sent
...	...	...	...	...	...	...

For a second example, again suppose that there is a constant frame rate with each frame needing distinct values for TopPicOrderCnt and BottomPicOrderCnt. Supp ose the frames are to be reordered from the display order IBBPBBP... to coding order IPBBPBB... with the first I frame an IDR frame, another I non -IDR frame every 15 frames, all P frames as references, and all B frames as non-references. In this case, mode 0 can be set up with offset_for_ref_frame[0] = 6, offset_for_non_ref_pic = -4, offset_for_top_to_bottom_field = 1, and pic_order_present_flag = 0. In this case,

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

OffsetPicOrderCnt will always correctly predict the picture order counts for the P frames an d half of the B frames. Table 8-2 illustrates how this works for the first few frames.

Table 8-2 – Second example for picture order count type 0

Frame Type	frame_num	FramePicOrderCnt	TopPicOrderCnt	BottomPicOrderCnt	delta_pic_order_cnt[0]	delta_pic_order_cnt[1]
I	0	0	0	1	0	0 not sent
P	1	6	6	7	0	0 not sent
B	2	2	2	3	0	0 not sent
B	2	4	4	5	2	0 not sent
P	2	12	12	13	0	0 not sent
B	3	8	8	9	0	0 not sent
B	3	10	10	11	2	0 not sent
...	...	...	...	...	...	...

Mode 0 can also code picture order counts when an encoder is reordering the pictures from display order to coding order in an irregular pattern. [Ed. Note: Consistency – Is this mode 0 or type 0?]

For an example of this, suppose tha t in the previous example some of the frames have duration 1.5 times the duration of the rest of the frames, and suppose that of the first seven frames the second B frame is the only longer duration frame. We can use the same setup as above except that pi c_order_present_flag = 1. Table 8-3 illustrates how this works for the first seven frames.

Table 8-3 – Third example for picture order count type 0 [Ed. Note: Consistency – Should this have a frame_num column?]

Frame Type	FramePicOrderCnt	TopPicOrderCnt	BottomPicOrderCnt	delta_pic_order_cnt[0]	delta_pic_order_cnt[1]
I	0	0	1	0	0
P	7	8	7	2	-2
B	2	2	3	0	0
B	4	4	5	2	0
P	13	14	13	0	-2
B	9	10	9	0	-2
B	11	12	11	2	-2
...	...	...	...	...	...

For an example of mode 1, suppose that there is a piecewise constant frame rate with a gap of one frame to be used perhaps by a systems layer for a frame repeat. Each frame needs distinct values for TopPicOrderCnt and BottomPicOrderCnt. Suppose the frames are to be reordered from the display order IBBPBBP... to coding order IPBBPBB... with the first I frame an IDR frame, and other I non -IDR frames every 15 frames directly following the one - frame gap, all P frames as references, and all B frames as non -references. In this case, mode 1 can be set up with num_ref_frames_in_pic_order_cnt_cycle = 5, offset_for_ref_frame = {6 ,6,6,6,8}, offset_for_non_ref_pic = -4, offset_for_top_to_bottom_field = 1, and pic_order_present_flag = 0 (i.e. Delta_pic_order_cnt[1] is always 0). Table 8-4 illustrates how this works for the first few frames. The mode 1 case is resilient to loss of non -IDR reference frames, while mode 0 is not. Both modes are resilient to loss of non -reference frames.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 8-4 – First example for picture order count type 1

Frame Type	frame_num	FramePicOrderCnt	TopPicOrderCnt	BottomPicOrderCnt	ExpectedPicOrderCnt	Delta_pic_order_cnt[0]
I	0	0	0	1	0	0
P	1	6	6	7	6	0
B	2	2	2	3	2	0
B	2	4	4	5	2	2
P	2	12	12	13	12	0
B	3	8	8	9	8	0
B	3	10	10	11	8	2
P	3	18	18	19	18	0
B	4	14	14	15	14	0
B	4	16	16	17	14	2
P	4	24	24	25	24	0
B	5	20	20	21	20	0
B	5	22	22	23	20	2
I	5	32	32	32	32	0
B	6	26	26	26	28	2
B	6	28	28	28	28	0
...	...	...	...	...	...	...

For a second example of mode 1 where the predictions don't work out as smoothly, suppose that there is a constant frame rate with each frame using the same values for TopPicOrderCnt and BottomPicOrderCnt. Suppose the frames are to be reordered from the display order IBBPBBP... to coding order IPBBPBB... with the first I frame an IDR frame, another I non-IDR frame every 15 frames, all P frames as references, and all B frames as references also. In this case, mode 1 can be set up with num_ref_frames_in_pic_order_cnt_cycle = 3, offset_for_ref_frame = {4, -2, 1}, offset_for_top_to_bottom_field = 0, and pic_order_present_flag = 0 (i.e. Delta_pic_order_cnt[1] is always 0). Table 8-5 illustrates how this works for the first few frames.

Table 8-5 – Second example for picture order count type 1

Frame Type	frame_num	FramePicOrderCnt	TopPicOrderCnt	BottomPicOrderCnt	ExpectedPicOrderCnt	delta_pic_order_cnt[0]
I	0	0	0	0	0	0
P	1	3	3	3	4	-1
B	2	1	1	1	2	-1
B	3	2	2	2	3	-1
P	4	6	6	6	7	-1
B	5	4	4	4	5	-1
B	6	5	5	5	6	-1
P	7	9	9	9	10	-1
B	8	7	7	7	8	-1
B	9	8	8	8	9	-1
P	10	12	12	12	13	-1
...	...	...	...	...	...	...

8.2.3 Decoding process for redundant slices

Input to this process is a coded slice.

Output of this process is a decoded approximation of the area of the primary picture represented in the redundant slice.

If the redundant_pic_cnt in the slice header of a coded slice is greater than 0, the decoder may discard the coded slice. If some of the samples in the decoded primary picture are incorrect and if the coded redundant slice can be correctly decoded, the decoder should replace the incorrect samples with the corresponding samples of the redundant decoded slice.

8.2.4 Decoding process for macroblock to slice group map

Input to this process are the active picture parameter set and the slice header of the slice to be decoded.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Output of this process is a macroblock to slice group map. [Ed.Note: Define variable ?]

Macroblock to slice group maps of mb_slice_group_map_type 0, 1, 2, and 6 are specified in the semantics of those types.

If decoding of a new picture is started as detected using the process in subclause 8.2.1 if num_slice_groups_minus1 is greater than 0 and if mb_slice_group_map_type is 3, 4, or 5, a macroblock to slice group map is generated. Slice group 0 has a growth order specified in subclauses 8.2.4.1-8.2.4.4. The number of macroblocks in slice group 0 is equal to slice_group_change_cycle × SliceGroupChangeRate. This number of macroblock locations in the specified growth order is allocated for slice group 0. The rest of the macroblocks of the picture are allocated for slice group 1.

8.2.4.1 Specification of allocation order for box-out for decoding process for macroblock to slice group map

Let H denote the number of coded macroblock rows of the picture and W denote the number of coded macroblocks columns of the picture. Macroblock locations are indicated with coordinates (x, y), where the top-left macroblock location of the picture has coordinates (0, 0) and the bottom-right macroblock location has coordinates (W − 1, H − 1). The mapping is created using a AllocationDirection variable that indicates the next macroblock location relative to the current one. AllocationDirection can have four values: (-1, 0), (1, 0), (0, -1) and (0, 1). Furthermore, the left-most and right-most macroblock columns in the mapping and the top-most and bottom-most macroblock rows in the mapping are stored in the variables Left, Right, Top, and Bottom respectively. For the box-out clockwise macroblock to slice group map type, the first macroblock location in the to slice group order is (x, y) = (W/2, H/2) and the initial AllocationDirection is (-1, 0). For the counter-clockwise macroblock to slice group map type, the first macroblock location in the mapping is (x, y) = ((W − 1)/2, (H − 1)/2) and the initial AllocationDirection is (0, 1). At the beginning, Left = Right = x, and Top = Bottom = y. A subsequent macroblock location (x, y) in the mapping is mapped by searching the first row from top to bottom in Table 8-6 for the same value of AllocationDirection and where the given condition is true. Then, the x, y, AllocationDirection, Left, Right, Top and Bottom variables are updated according to the refined macroblock to slice group map type. If Left >= 0, Right < W, Top >= 0 and Bottom < H, the next macroblock location (x, y) in the mapping is mapped as described above. Otherwise, all macroblock locations have been mapped to slice groups.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 8-6 – Mapping for the box-out macroblock map to slice group type

AllocationDirection	Condition	Box-out clockwise	Box-out counter-clockwise
(-1, 0)	x > Left	x = x – 1	x = x – 1
(-1, 0)	x = = 0	y = Top – 1 Top = Top – 1 AllocationDirection = (1, 0)	y = Bottom + 1 Bottom = Bottom + 1 AllocationDirection = (1, 0)
(-1, 0)	x = = Left	x = x – 1 Left = Left – 1 AllocationDirection = (0, -1)	x = x –1 Left = Left – 1 AllocationDirection = (0, 1)
(1, 0)	x < Right	x = x + 1	x = x + 1
(1, 0)	x = = W – 1	y = Bottom + 1 Bottom = Bottom + 1 AllocationDirection = (-1, 0)	y = Top – 1 Top = Top – 1 AllocationDirection = (-1, 0)
(1, 0)	x = = Right	x = x + 1 Right = Right + 1 AllocationDirection = (0, 1)	x = x + 1 Right = Right + 1 AllocationDirection = (0, -1)
(0, -1)	y > Top	y = y – 1	y = y – 1
(0, -1)	y = = 0	x = Right + 1 Right = Right + 1 AllocationDirection = (0, 1)	x = Left – 1 Left = Left – 1 AllocationDirection = (0, 1)
(0, -1)	y = = Top	y = y – 1 Top = Top –1 AllocationDirection = (1, 0)	y = y –1 Top = Top –1 AllocationDirection = (-1, 0)
(0, 1)	y < Bottom	y = y + 1	y = y + 1
(0, 1)	y = = H – 1	x = Left – 1 Left = Left – 1 AllocationDirection = (0, -1)	x = Right + 1 Right = Right + 1 AllocationDirection = (0, -1)
(0, 1)	y = = Bottom	y = y + 1 Bottom = Bottom + 1 AllocationDirection = (-1 ,0)	y = y + 1 Bottom = Bottom + 1 AllocationDirection = (1 ,0)

8.2.4.2 Specification of the mapping for raster scan for decoding process for macroblock to slice group map

For the raster scan slice group macroblock to slice group map type, the first macroblock in the mapping is the top -left one of the picture, and the mapping follows the raster scan order.

For the reverse raster scan slice group macroblock to slice group map type, the first macroblock in the mapping is the bottom-right one of the picture, and the mapping follows the reverse raster scan order.

8.2.4.3 Specification of allocation order for wipe for decoding process for macroblock to slice group map

For the wipe right slice group macroblock to slice group map type, the first macroblock in the mapping is the top -left one of the picture. The mapping runs from top to bottom. The next macroblock after the b ottom macroblock of a column is the top macroblock of the column to the right of the previous column.

For the wipe left slice group macroblock to slice group map type, the first macroblock in the mapping is the bottom -right one of the picture. The mapping runs from bottom to top. The next macroblock after the top macroblock of a column is the bottom macroblock of the column to the left of the previous column.

8.2.4.4 Specification of allocation order for macroblock level adaptive frame and field coding for decoding process for macroblock to slice group map

[Ed. Note: Very large value in Format → Paragraph → Special → Hanging Indent for the above style.]

Mapping follows **Error! Reference source not found.** [Ed. Note: missing cross-reference]in subclause 6.3, instead of raster scan. [Ed.Note: more text need here !!!]

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

8.2.5 Decoding process for slice slice data partitioning

[Ed.Note: consider to move this to clause 7]

Input to this process are slice data partition A layer RBSP, slice data partition B layer RBSP, and slice data partition C layer RBSP that have the same slice_id and redundant_pic_cnt. Slice data partition B layer RBSP and slice data partition C layer RBSP may not be present.

Output of this process is a coded slice.

When slice slice data partitioning is not used, coded slices start with a slice header and are followed by all the entropy coded symbols of Categories 4, 5, and 6 (see Category column in clause 6.4) of the macroblock data for the macroblocks of the slice.

When slice data partitioning is used, the macroblock data of a slice is partitioned in three partitions contained in 3 or more NAL units [Ed. Note: More? I don't think so.]. Partition A contains a partition A header and all syntax elements of Category 4. Partition B contains a partition B header and all syntax elements of Category 5. Partition C contains a partition C header and all syntax elements of Category 6.

When slice data partitioning is used, each partition is parsed from a separate NAL unit, which may be empty if no symbols of the respective Category exist. The decoder shall process the three partitions of a coded slice equally to a corresponding slice layer RBSP by multiplexing between them depending on the partition assignment in the syntax tables in subclause 7.3. [Ed. Note: be more specific about how the decoder parses categories 5 and 6]

NOTE - Symbols of Category 5 are relevant to decoding of I and SI macroblock types. Symbols of Category 6 are relevant to the decoding of residual data of P and B macroblock types. Category 4 encompasses all other symbol types related to the decoding of macroblocks, and their information is often denoted as header information. The Partition A header contains all the symbols of the slice header, and additionally a slice number that is used to associate the partitions B and C with the partition A. The partition B and C headers contain only the slice number which allows their association with the partition A of the slice

8.2.6 Decoding process for reference picture reordering

Input to this process are the syntax elements for the reference index reordering as specified in subclause 7.3.3.1.

Output of this process are reference index list 0 and reference index list 1.

The decoded picture buffer contains frame stores. Each frame store may contain one field or a pair of complimentary fields. Complimentary fields have the same frame number but opposite parity. A frame store is in use when it contains at least one field. The decoder stores decoded pictures in the decoded picture buffer as indicated by the bitstream. Pictures are stored for prediction of pictures that are subsequent to the current picture in decoding order and/or because their decoded picture output time is later than their decode time. A picture that is stored for prediction, referred to as a reference picture, is marked as "used for reference", and is labelled as a short-term reference picture, in whicb case it is identified by its frame_number and parity. A short-term picture may be assigned a long-term reference frame index, in which case it is identified by its long-term index. A picture that is not stored for prediction is labelled as "unused for prediction". Eacb stored picture is associated with an output time indication. Pictures are removed from the decoded frame buffer when they are marked as "unused for reference" and their output time indication is earlier than or equal to the current time. Short-term reference pictures can remain marked for reference for a finite duration, given by MaxFrameNum. Long term reference pictures can remain marked "used for reference" until the next IDR picture. Memory_management_control_operation commands are used to modify the contents of the buffer.

Reference pictures are addressed through reference indices. A reference index is an index into a list of reference pictures stored in the decoded picture buffer. Wben decoding a P or SP slice there is a single list (list 0) of the reference picture indices. When decoding a B slice there is a second list (list 1) in addition to list 0 that lists the reference pictures in an order that is independent of the ordering of list 0. Prediction of a block in a B slice may be based on pictures from one or both lists.

At the beginning of decoding of each slice, the default index order is created using picture numbers and long-term indices. This default index order is mapped to an index order according to the remapping_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_idx fields. The decoder employs these (remapped) indices when referencing a picture for inter prediction on the macroblock layer. The total length of the remapped list is num_ref_idx_l0_active_minus1+1 for list 0 and is num_ref_idx_l1_active_minus1+1 for list 1. A given reference picture may appear at more than one index position in the remapped list 0 or list 1.

8.2.6.1 Decoding process for picture numbers

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

Picture numbers are used to index reference pictures within the decoded picture buffer in the decoding process and for management and of the reference pictures for both changing the default indices and for controlling the contents of the

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

decoded picture buffer using memory management control operations. The construction of picture numbers depends upon the picture coding type of the picture being decoded. The picture numbers whe n decoding a coded frame are constructed for each entire short-term reference frame stored in the decoded picture buffer buffer. The picture number of a frame that has frame number FN is given by PN=FN.

The picture numbers when decoding a coded field are constructed for each short-term reference field stored in the decoded picture buffer. The picture number, PN, of a field that has frame number FN is given by PN=2 x FN if the field is a top field, and it is given by PN = 2 x FN+1 if the field is a bottom field.

Long-term picture indices are also used in the decoding process for management and use of the decoded frame buffer. Long-term picture indices are used for assigning a long-term index to a short-term reference picture, marking a long-term reference picture as "unused for reference", and for changing the default indices of pictures that have been assigned a long-term index.

If the current decoded picture is a frame, the long-term picture number, LTPN, of a frame that has long-term frame index LTFI, is given by LTPN = LTFI.

If the current decoded picture is a field, the long-term picture number, LTPN, of a field that has long-term frame index LTFI, is given by LTPN = 2 x LTFI if the field is a top field, and is given by LTPN = 2 x LTFI + 1 if the field is a bottom field. [Ed.Note: rewrite top and bottom field to parities]

If the current picture being decoded is a frame, the parameter MaxPicNum is specified to equal MaxFrameNum, and if the current picture being decoded is a field-structured picture, the parameter MaxPicNum is specified to equal 2 MaxFrameNum.

8.2.6.2 Decoding process for frame numbers

Input to this process are the frame number of the previous reference picture in decoding order and the frame number of the reference picture to be decoded.

Output of this process is decoded picture buffer management operations.

This process is performed when decoding of a new reference picture is started from the slice to be decoded.

Let PrevFrameNum be the frame number of the previous reference picture in decoding order and CurrFrameNum be the frame number of the slice to be decoded.

The decoder shall infer a gap in frame numbers if both of the following conditions are true:

1. The slice to be decoded does not belong to an IDR picture
2. (PrevFrameNum + 1) % MaxFrameNum != CurrFrameNum

If a gap in frame number is inferred, the set of missing frame numbers is calculated. If CurrFrameNum > PrevFrameNum, the set of missing frame numbers ranges from PrevFrameNum + 1 to CurrFrameNum – 1, inclusive. If CurrFrameNum <= PrevFrameNum, the set of missing frame numbers ranges from PrevFrameNum + 1 to MaxFrameNum – 1, inclusive, and from 0 to CurrFrameNum – 1, inclusive. [Ed.Note: this is normative ... what about weird cases ?]

If a gap in frame numbers is inferred and if required_frame _num_update_behaviour_flag equals 1, the decoder shall act as if reference frames corresponding to the set of missing frame numbers had been inserted into the decoded picture buffer using the sliding window reference picture buffering mode. [Ed. Note: Try to come up with more precise wording than "act as if..."] An index for a missing frame number in the decoded picture buffer is called an "invalid" index. The decoder should infer an unintentional picture loss if any "invalid" index is referred to in inter pr ediction or if an "invalid" index is re-mapped. The decoder should not infer an unintentional picture loss if a memory management control operation is applied to a frame having an "invalid" index. [Ed.Note: "invalid" ? change.]

If required_frame_num_update _behaviour_flag is equal to 0, the decoder should infer an unintentional picture loss if a gap in frame numbers is inferred or if a picture not stored in the decoded picture buffer is indicated in an abs_diff_pic_num_minus1 or long_term_pic_idx field.

NOTE - In case of an unintentional picture loss, the decoder may invoke some concealment process. If required_frame_num_update_behaviour_flag is equal to 1, the decoder may replace the picture corresponding to an "invalid" index with an error-concealed one and remove the "invalid" indication. Otherwise, the decoder may insert an error-concealed picture into the decoded picture buffer assuming the sliding window reference picture buffering mode. Concealment may be conducted by copying the previous decoded picture in decoding order into the position of the missing picture. In addition or instead, the decoder may send a forced intra update signal to the encoder by external means (for example, Recommendation H.245) if such external means is available, or the decoder may use external means or back-channel messages (for example, Recommendation H.245) to indicate the loss of pictures to the encoder if such external means is available.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

8.2.6.3 Initialization process for reference indices

This intitialization process is invoked when ... [Ed.Note: specify]

Output of this process is/are ??? [Ed.Note: specify output of this process]

A reference index is a relative index into a list of reference indices to indicate which reference picture out of the decoded picture buffer is used for inter prediction. When decoding a P or SP slice, there is one such list of reference indices, called the reference index list 0. When decoding a B slice, there may be two reference indices used per block each pointing into separate lists of reference indices which are called the reference index list 0 and reference index list 1.

Reference index list 0 and reference index list 1 have default mappings to the pictures numbers in the decoded picture buffer as specified below.

Default index order for P and SP slices

8.2.6.3.1 Initialization process for reference indices for P and SP slices in frames

This intitialization process is invoked when ... [Ed.Note: specify]

Output of this process is/are ??? [Ed.Note: specify output of this process]

When decoding a frame, the default index order for list 0 prediction of P and SP slices in frames is for the short-term reference frames to precede the long-term reference frames in the reference indexing order. Within the set of short-term reference frames, the default order is for the frames to be ordered starting with the most recently-decoded frame and proceeding through to the reference frame in the decoded picture buffer that was decoded first (i.e., in decreasing order of frame_num in the absence of wrapping of the frame_num value). Within the set of long-term frames, the default order is for the frames to be ordered starting with the frame with the smallest long-term index and proceeding to the frame with largest long-term index.

A field that is stored in the decoded picture buffer and is marked as "used for reference" for which the opposite parity field is not stored in the decoded picture buffer or marked as "unused for reference" shall not be included in the default index order, shall not be remapped, and shall not be used for prediction in frames.

For example, assuming no wrap of the frame_num field, if the decoded picture buffer contains three short-term reference frames with frame_num equal to 300, 302, and 303 and two long-term reference frames with long-term frame indices 0 and 3, all being marked as "used for reference", the default index order is:

- default index 0 refers to the short-term reference frame with frame_num 303,
- default index 1 refers to the short-term reference frame with frame_num 302,
- default index 2 refers to the short-term reference frame with frame_num 300,
- default index 3 refers to the long-term reference frame with long-term frame index 0, and
- default index 4 refers to the long-term reference frame with long-term frame index 3.

8.2.6.3.2 Initialization process for reference indices for P and SP slices in fields

This intitialization process is invoked when ... [Ed.Note: specify]

Output of this process is/are ??? [Ed.Note: specify output of this process]

When decoding a field, each field of the reference frames is identified as a separate reference picture with a unique index. Thus fields effectively have at least twice the number of pictures available for referencing.

The default index order for list 0 prediction of P and SP slices in fields is for the short-term reference fields to precede the long-term reference fields in the reference indexing order.

Within the set of short-term reference fields, the default order is calculated by following Step P_list0_1a and Step P_list0_1b.

Step P_list0_1a:

An ordered list of frames is created by ordering the short-term reference frames as in the case of creating the default index order for P and SP slices in frames, except that the current frame shall be included at the start of the list when the current field is the second of a frame and the first field of the frame is stored and marked "used for reference", and except that frames in which only one field is stored and marked "used for reference" are included in the list.

Step P_list0_1b:

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

The default index order is created by selecting reference fields from this ordered list of frames from step P_list0_1a by alternating between fields of the same parity and fields of the opposite parity, starting with fields that have the same parity as the current field. If one field of a reference frame was not decoded or not stored or stored but marked "unused for reference", the index order calculation shall ignore the missing field and instead index the next available stored reference field of the respective parity from the ordered list of frames. When there are no more short-term reference fields of the respective parity in the ordered list of frames, default indices shall be allocated to the not yet indexed fields of the other parity in the order in which they occur in the ordered list of frames.

Within the set of long-term fields, the default order is calculated by following Step P_list0_2a and Step P_list0_2b.

Step P_list0_2a:

An ordered list of frames is created by ordering the long-term reference frames starting with the frame with the smallest long-term index and proceeding to the frame with largest long-term index.

Step P_list0_2b:

The default index order is created by selecting reference fields from this ordered list of frames from step P_list0_2a using the procedure of Step P_list0_1b.

Figure 8-16 shows the case of the first field in a field-structured picture pair, while Figure 8-2 shows the case of the second field when no long-term fields are stored for reference.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

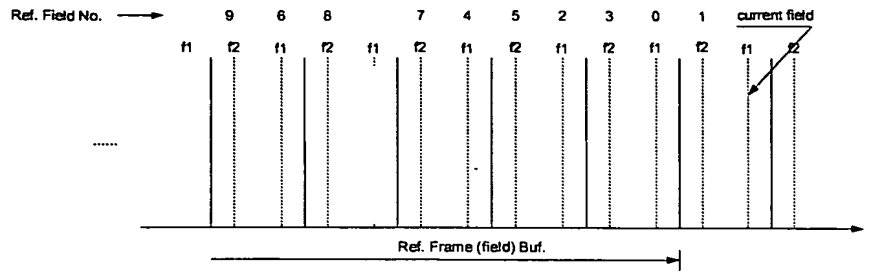


Figure 8-1 – Default reference field number assignment when the current picture is the first field coded in a frame [Ed. Note: Need to change f1/f2 to f0/f1 – always use zero-based indexing]

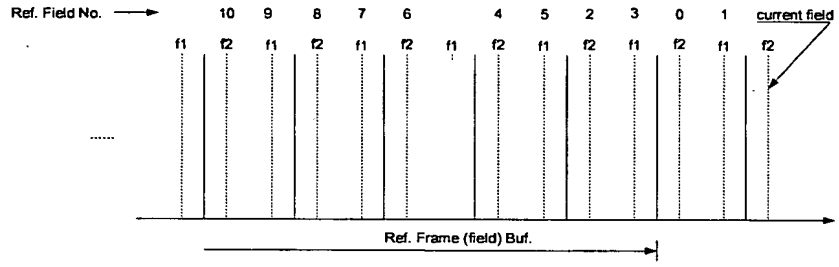


Figure 8-2 – Default reference field number assignment when the current picture is the second field coded in a frame [Ed. Note: Need to change f1/f2 to f0/f1 – always use zero-based indexing]

8.2.6.3.3 Initialization process for reference indices for B slices in frames

This intitialization process is invoked when … [Ed.Note: specify]

Output of this process is/are ??? [Ed.Note: specify output of this process]

The organisation of short-term reference pictures in the default order for B slices depends on output order, as given by PicOrderCnt. In the case of a reference frame having being coded as two fields with different values of PicOrderCnt, the frame shall be considered to have the smaller of these two values of PicOrde rCnt for the purpose of constructing the default index order for B slices in frames.

When decoding a frame, the default index order for list 0 prediction of B slices is for the short -term reference frames (i.e., frames which have not been given a long -term index) to precede the long-term reference frames in the reference indexing order. Within the set of short-term reference frames, the default order is for the frames to be ordered starting with the frame with the largest value of PicOrderCnt less than the value of PicOrderCnt of the current frame and proceeding through to the short -term reference frame that has the smallest value of PicOrderCnt; and then the frame with the smallest value of PicOrderCnt greater than the value of PicOrderCnt of the current f rame and proceeding through to the short-term reference frame that has the largest value of PicOrderCnt. Within the set of long -term reference frames, the default order is for the frames to be ordered starting with the frame with the smallest long -term index and proceeding to the frame with the largest long-term index.

When decoding a frame, the default index order for list 1 prediction of B slices is for the short -term reference frames to precede the long-term reference frames in the reference indexing ord er. Within the set of short-term reference frames, the default order is for the frames to be ordered starting with the frame with the smallest value of PicOrderCnt greater than the value of PicOrderCnt of the current frame and proceeding through to the sh ort-term reference frame that has the

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

largest value of PicOrderCnt; and then the frame with the largest value of PicOrderCnt less than the value of PicOrderCnt of the current frame and proceeding through to the short -term reference frame that has the smallest value of PicOrderCnt . Within the set of long-term reference frames, the default order is for the frames to be ordered starting with the frame with the smallest long-term index and proceeding to the frame with the largest long -term index.

The ordinary default order specified in the previous paragraph shall be used as the default index order for list 1 prediction unless there is more than one reference picture in the set and the ordinary default index order for list 1 prediction is the same as the default index order for list 0 prediction. In this exceptional case, the default index order for list 1 prediction shall be the ordinary default index order with the order of the first two pictures switched.

A field that is stored in the decoded picture buffer and is marked as "used for reference" for which the opposite parity field is not stored in the decoded picture buffer or is stored but marked as "unused for reference" shall not be included in the default index order, shall not be remapped, and shall not b e used for prediction in frames.

8.2.6.3.4 Initialization process for reference indices for B slices in fields

This intitialization process is invoked when ... [Ed.Note: specify]

Output of this process is/are ??? [Ed.Note: specify output of this process]

When decoding a field, each field of the stored reference frames is identified as a separate reference picture with a unique index. Thus field structured pictures effectively have at least twice the number of pictures available for referencing.

The default index order for list 0 prediction of B slices in fields is for the short -term reference fields to precede the long-term reference fields in the reference indexing order.

Within the set of short-term reference fields, the default order list 0 prediction is calculated b y following Step B_list0_1a and Step B_list0_1b.

Step B_list0_1a:

An ordered list of frames is created by ordering the short -term reference frames as in the case of creating the default list 0 index order for B slices in frames, except that the current fr ame shall be included at the start of the list when the current field is the second of a frame and the first field of the frame is stored and marked "used for reference", and except that frames in which only one field is stored and marked "used for referen ce" are included in the list.

Step B_list0_1b:

The default index order is created by selecting reference fields from this ordered list of frames from step B_list0_1a using the procedure of Step P_list0_1b.

Within the set of long-term fields, the default order is calculated by following Step B_list0_2a and Step B_list0_2b.

Step B_list0_2a:

An ordered list of frames is created by ordering the long -term reference frames starting with the frame with the smallest long-term index and proceeding to the frame with largest long-term index.

Step B_list0_2b:

The default index order is created by selecting reference fields from this ordered list of frames from step B_list0_2a using the procedure of Step P_list0_1b.

The default index order for list 1 prediction of B sli ces in fields is for the short-term reference fields to precede the long-term reference fields in the reference indexing order.

Within the set of short-term reference fields, the default order for list 1 prediction is calculated by following Step B_list1_1a and Step B_list1_1b.

Step B_list1_1a:

An ordered list of frames is created by ordering the short -term reference frames as in the case of creating the default list 1 index order for B slices in frames, except that the current frame shall be included imme diately before the frame with the largest value of PicOrderCnt less than the value of PicOrderCnt of the current frame when the current field is the second of a frame and the first field of the frame is stored and marked "used for reference", and except th at frames in which only one field is stored and marked "used for reference" are included in the list.

Step B_list1_1b:

The default index order is created by selecting reference fields from this ordered list of frames from step B_list1_1a using the procedu re of Step P_list0_1b.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Within the set of long-term fields, the default order is calculated by following Step B_list1_2a and Step B_list1_2b.

Step B_list1_2a:

An ordered list of frames is created by ordering the long -term reference frames starting with the frame with the smallest long-term index and proceeding to the frame with largest long -term index.

Step B_list1_2b:

The default index order is created by selecting reference fields from this ordered list of frames from step B_list1_2a using the procedure of Step P_list1_1b.

8.2.6.4 Assignment process for reference indices

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

The syntax elements remapping_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_idx fields allow indexing into the decoded picture buffer to be temporarily altered from the default index order for the decoding of the current slice. A remapping_of_pic_nums_idc "end loop" indication indicates t he end of a list of reordering commands.

The indices are assigned starting at zero and increasing by one for each remapping_of_pic_nums_idc field. Pictures that are not re-mapped to a specific order by remapping_of_pic_nums_idc, shall follow after any pict ures having a re-mapped order in the indexing scheme, following the default order amongst these non -re-mapped pictures.

8.2.6.4.1 Assignment process for reference indices for short-term pictures

Input to this process is/are ??? [Ed.Note: specify input to this proces s]

Output of this process is/are ??? [Ed.Note: specify output of this process]

abs_diff_pic_num_minus1 plus one indicates the absolute difference between the picture number of the picture being re - mapped and the prediction value. For the first occurence of the abs_diff_pic_num_minus1 field in ref_idx_reordering( ), the prediction value is the picture number of the current picture. For subsequent occurences of the abs_diff_pic_num_minus1 field in ref_idx_reordering( ), the prediction value is the picture num ber of the picture that was re-mapped most recently using abs_diff_pic_num_minus1.

The decoder shall determine the picture number of the picture being re -mapped, PNQ, in a manner mathematically equivalent to the following, where the picture number predicti on is PNP.

```
if( remapping_of_pic_nums_idc = = 0 )     /* a negative difference */
  if( PNP - abs_diff_pic_num_minus1 < 0 )
    PNQ = PNP - abs_diff_pic_num_minus1 + MaxPicNum - 1;
  else
    PNQ = PNP - abs_diff_pic_num_minus1;
else   /* a positive difference */
  if( PNP + abs_diff_pic_num_minus1 > MaxPicNum-1 )
    PNQ  = PNP + abs_diff_pic_num_minus1 - MaxPicNum + 1;
  else
    PNQ  = PNP + abs_diff_pic_num_minus1 + 1;
```

The encoder shall control remapping_of_pic_nums_idc and abs_diff_pic_num_minus1 such that t he decoded value of abs_diff_pic_num_minus1 shall not be greater than or equal to MaxPicNum.

As an example implementation, the encoder may use the following process to determine values of abs_diff_pic_num_minus1 and remapping_of_pic_nums_idc to specify a r e-mapped picture number in question, PN:

```
DELTA = PNQ - PNP;
if( DELTA < 0 )
    if( DELTA < -MaxPicNum/2 - 1 )
        MDELTA = DELTA + MaxPicNum;
    else
        MDELTA = DELTA;
else
```

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

```
        if ( DELTA > MaxPicNum/2 )
            MDELTA = DELTA - MaxPicNum;
        else
            MDELTA = DELTA;
    abs_diff_pic_num_minus1 = Abs( MDELTA ) - 1;
```

remapping_of_pic_nums_idc is then determined by the sign of MDELTA.

The prediction value used by any subsequent abs_diff_pic_num_minus1 re -mappings is not affected by long_term_pic_idx.

8.2.6.4.2 Assignment process for reference indices for long-term pictures

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

The long_term_pic_idx field indicates the index of the long -term reference picture being remapped.

8.2.7 Decoded picture buffer management process

Input to this process are syntax elements for the decoded picture buffer management 7.3.3.3 and the decoded picture buffer.

Output of this process is the possibl y modified decoded picture buffer.

A picture that is stored for prediction, referred to as a reference picture, is marked as "used for reference", and is labelled as either a short-term reference picture, in which case it is identified by its frame_number and parity, or is labelled as a long-term reference picture, in which case it is identified by its long -term index and parity. A picture that is not stored for prediction is labelled as "unused for reference".

The decoded frame buffer is initially empty, a nd the value of max_long_term_pic_idx_plus1 shall initially be zero.

When nal_ref_idc is not equal to zero, the current picture is stored in the decoded picture buffer and is marked "used for reference".

The decoded picture buffer is modified as indicated by adaptive_ref_pic_buffering_mode_flag, which indicates either "Sliding Window", a first-in, first-out mechanism, or "Adaptive Memory Control", a customised adaptive buffering operation specified with memory_management_control_operation commands. A pictur e that is stored in the decoded picture buffer and marked "used for reference" can be used as a reference for prediction until it is marked as "unused for reference" by either the "Sliding Window" or "Adaptive Memory Control" mechanism, after which it can not be used as a reference for prediction.

In frames, memory_management_control_operation commands apply to both fields of the frame indicated.

In fields, memory_management_control_operation commands apply to individual fields indicated.

The decoded picture buffer is a buffer that contains frames. Each frame in the buffer may contain a top field, a bottom field, or a top and bottom field, but in the latter case, both must have the same frame_num. When decoding a field, if a memory_management_control_operati on command assigns a long -term index to a field, another memory_management_control_operation command to assign the same long -term index to the other field of the same frame shall be present in the same dec_pic_buffer_management( ) if that field is stored and marked as "used for reference".

8.2.7.1 Sequence of operations for decoded picture buffer management process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

The sequence of operations for the management of the decoded picture buffer is as follows.

Step 1:

The picture is decoded.

Step 2:

The decoded picture buffer is updated according to either the "Sliding Window" or "Adaptive Memory Control" mechanism as indicated by adaptive_ref_pic_buffering_mode_flag. In the case of "Adaptive Memory Control", the memory_management_control_operation commands are applied in the order in which they occur in the bitstream.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

If the picture is an IDR picture, all reference pictures are marked as "unused for reference" before performing the "Sliding Window" mechanism and before processing any memory_management_control_operation commands.

Step 3:

If nal_ref_idc indicates that the picture is to be stored for reference, the picture is s tored in the decoded picture buffer and is marked "used for reference". If the current picture is a frame or the first field of a frame marked "used for reference", there shall be a completely empty reference frame memory in the buffer into which the pictu re can be stored. If the current picture is the second stored reference field of a frame, it shall be stored in the same frame as the first field of the frame.

After adding a reference picture to the decoded picture buffer, the number of frames marked as " used for reference", including frames in which only one field is marked as "used for reference" shall not be greater than num_ref_frames.

8.2.7.2 Sliding window decoded picture buffer management process

Input to this process is/are ??? [Ed.Note: specify input to t his process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

The sliding window reference picture buffering mode operates as specified in this paragraph, when nal_ref_idc indicates that the current picture is to be stored as a reference picture and the adaptive_ref_pic_buffering_mode_flag indicates "Sliding Window". The syntax element num_ref_frames of the sequence parameter set is used to define the operation of the sliding window reference picture buffering mode.

The sliding window reference picture buffering mode actions are only applied to the decoded picture buffer on processing of the first stored reference field of a frame. No action is taken when processing the second stored reference field of a frame.

Set S equal to the number of short-term reference frames in the decoded frame buffer that are marked as "used for reference", including short-term reference frames in which only one field is marked as "used for reference". Set L equal to the number of long-term reference frames in the decoded frame buffer that are marked as "used for reference", including long-term reference frames in which only one field is marked as "used for reference". Set N to the total number of frames stored in the decoded picture buffer.

If S + L equals num_ref_frames, S shall be greater than zero, and the short -term reference frame that was decoded first (i.e. in the absence of wrapping of the frame_num value, the frame with the smallest value of frame_num) is marked as "unused for reference".

If N equals DPBcapacity, S shall be greater than zero, and the short -term reference frame that was decoded first is marked as "unused for reference".

Otherwise no action is taken.

NOTE - num_ref_frames may be less than DPBcapacity to allow specific space in the decoded picture buffer for frames that are marked as "unused for reference". However, the number of frames that are marked as "unused for reference" may be greater than DPBcapacity - num_ref_frames, but in this case it should be noted that it may not be possible to discard non stored pictures without attempting to decode them, as they may interact with the operation of the sliding window.

8.2.7.3 Adaptive Memory Control decoded picture buffer management process

Input to this process is/are ??? [Ed.Note: specify inpu t to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

The adaptive reference picture buffering mode allows all or specified pictures to be marked "unused for reference", allows specified pictures to be allocated a long-term frame index, allows the max_long_term_pic_idx_plus1 to be modified, and allows the whole buffer to be emptied, by use of memory_management_control_operation commands.

The element max_long_term_pic_idx_plus1 places a limit on the number of long -term frame indexes that may be used.

memory_management_control_operation commands are processed in the order they occur in the bitstream, and are processed after the whole picture has been decoded. When all commands have been processed, storage of the current picture is considered. When nal_ref_idc is not equal to 0, the current picture is marked "used for reference" and added to the decoded picture buffer initially as a short-term reference picture. memory_management_control_operation commands in the bitstre am shall be such that when nal_ref_idc indicates that the current picture is to be stored as a reference picture, there shall be sufficient capacity in the decoded picture buffer to store the current picture.

8.2.7.3.1 Assignment process for short-term pictures to "unused for reference"

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

If memory_management_control_operation equals 1 (Mark a Short -Term Reference Picture as "Unused for reference"), a specified short-term reference picture in the decoded picture buffer is marked as "unused for reference", if that picture has not already been marked as "unused for reference".

If the current decoded picture number is PNC, di fference_of_pic_nums_minus1 is used in an operation mathematically equivalent to the following equations, to calculate, PNQ, the picture number of the short -term reference picture to be marked as "unused for reference".

```
if(PNC < difference_of_pic_nums_minus1)
  PNQ = PNC - difference_of_pic_nums_minus1 - 1 + MaxPicNum;
else
  PNQ = PNC - difference_of_pic_nums_minus1 - 1;
```

Similarly, the encoder may compute the difference_of_pic_nums_minus1 value to encode using the following relation:

```
if(PNC < PNQ)
  difference_of_pic_nums_minus1 = PNC - PNQ - 1 + MaxPicNum;
else
  difference_of_pic_nums_minus1 = PNC - PNQ - 1;
```

8.2.7.3.2 Assignment process for long-term pictures to "unused for reference"

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If memory_management_control_operation equals 2 (Mark a Long -Term Reference Picture as "Unused for reference"), a specified long-term reference picture is marked as "unused for reference" , if that picture has not already been marked as "unused for reference".

The field long_term_pic_idx indicates the the long -term picture number, LTPN, of the picture to be marked as "unused for reference".

NOTE: this use of long_term_pic_idx is different t o its use when allocating a long-term index to a short-term reference picture.

8.2.7.3.3 Assignment process for a long-term index to a short-term reference picture

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If memory_management_control_operation equals 3 (Assign a Long -Term Index to a Picture), a specified short-term reference picture is allocated a specified long-term index, if that picture has not already been allocated a long-term index. If the picture specified in a long-term assignment operation is already associated with the required long_term_pic_idx, no action shall be taken by the decoder. The specified short -term reference picture is no longer considere d as a short-term reference picture following the processing of this command, and shall not be referenced at a later point in the bitstream by reference to its picture number, except for a repetition of the long -term index assignment.

If another picture is already assigned the same long-term index as the specified long-term index, the other picture is marked as "unused for reference".

The short-term reference picture to be assigned the long-term index is identified by its picture number, which is derived from difference_of_pic_nums_minus1 as in subclause 8.2.7.3.1.

When decoding a field, if a memory_management_control_operation command assigns a long -term index to a field, another memory_management_control_operation command to as sign the same long-term index to the other field of the same frame shall be present in the same dec_pic_buffer_management( ) if that field is stored in the decoded picture buffer and marked as "used for prediction". The long -term frame number of the frame into which the short-term picture is transferred is given by long_term_pic_idx.

long_term_pic_idx shall not be greater than max_long_term_pic_idx_plus1 −1.

The bitstream may contain the same long-term index assignment operation or max_long_term_pic_idx_plu s1 specification message repeatedly.

NOTE - If long_term_pic_idx does not satisfy this constraint, this condition should be treated by the decoder as an error. For error resilience, a repeated assignment of the long-term index or max_long_term_pic_idx_plus1 maybe desirable.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

A bitstream shall not assign a long-term index to a short-term reference picture that has been marked as "unused for reference" by the decoding process prior to the first such assignment message in the bitstream. A bitstream shall not assign a long-term index to a picture number that has not been included earlier in the bitstream.

Once a long-term picture index has been assigned to a picture, the only potential subsequent use of the long -term picture's picture number within the bitstream shall be in a repetition of the long-term index assignment. long_term_pic_idx becomes the unique ID for the life of a long -term picture.

8.2.7.3.4 Decoding process for maximum long-term index

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If memory_management_control_operation equals 4 (Specify the Maximum Long -Term Frame Index), max_long_term_pic_idx_plus1 indicates the maximum index allowed for long -term reference frames (until receipt of another value of max_long_term_pic_idx_plus1).

If max_long_term_pic_idx_plus1 is smaller than its previous value, all frames in the decoded picture buffer having indices greater than max_long_term_pic_idx_plus1 – 1 shall be marked "unused for reference".

If max_long_term_pic_idx_plus1 is greater than its previous value there are no immediate consequences on the decoded picture buffer.

NOTE - max_long_term_pic_idx_plus1 can therefore be used to mark long-term reference pictures as "unused for reference" but has no effect on short-term reference pictures. The frequency of transmitting max_long_term_pic_idx_plus1 is out of the scope of this Recommendation. However, the encoder should send a max_long_term_pic_idx_plus1 parameter upon receiving an error message, such as an intra request message.

8.2.7.3.5 Assignment process for all reference pictures to "unused for reference"

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If memory_management_control_operation equals 5 (reset all reference pictures), or the current picture is an IDR picture and the non_ref_pic_reset_flag is equal to 0, all reference pictures are removed from the decoded picture buf fer, and max_long_term_pic_idx_plus1 is set to zero. If memory_management_control_operation equals 5 (Reset all reference pictures), the picture shall be considered to have frame_num equal to 0 immediately after processing the memory_management_control_ope ration equal to 5.

8.2.7.3.6 Assignment process for all non-reference pictures to "unused"

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If memory_management_cont rol_operation equals 6 (reset all non -reference pictures) all non-reference pictures are removed from the decoded picture buffer.

8.2.7.3.7 Assignment process for all reference and non-reference pictures to "unused"

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If memory_management_control_operation equals 7 (reset all reference and non -reference pictures), or the current picture is an IDR picture and the non_ref_p ic_reset_flag is equal to 1, all reference and non-reference pictures are removed from the decoded picture buffer. If memory_management_control_operation equals 7 (reset all reference and non-reference pictures), the picture shall be considered to have fr ame_num equal to 0 immediately after processing the memory_management_control_operation equal to 7.

8.2.8 Decoding process for macroblock level frame/field adaptive coding

[Ed.Note: this should go somewhere else in the spec … to subclause 8.5.3]

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

When mb_ frame_field_adaptive_flag is equal to 1, the decoded frame is scanned on a macroblock pair by macroblock pair basis, as shown in Figure 6-4. A macroblock pair can be decoded in either frame or field decoding mode. For frame decoding mode, a macroblock pair is decoded as two frame macroblocks, and each can be further divided into one of

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

block patterns shown in Figure 6-5. For field decoding mode, a macroblock pair is first split into one top -field macroblock and one bottom -field macroblock, as shown in Figure 8-3. The top-field macroblock and the bottom -field macroblock are further divided into block one of patterns shown in **Error! Reference source not found.**. Each macroblock in either frame or field decoding mode can have a different mb_type described in subclause 7.4.5.

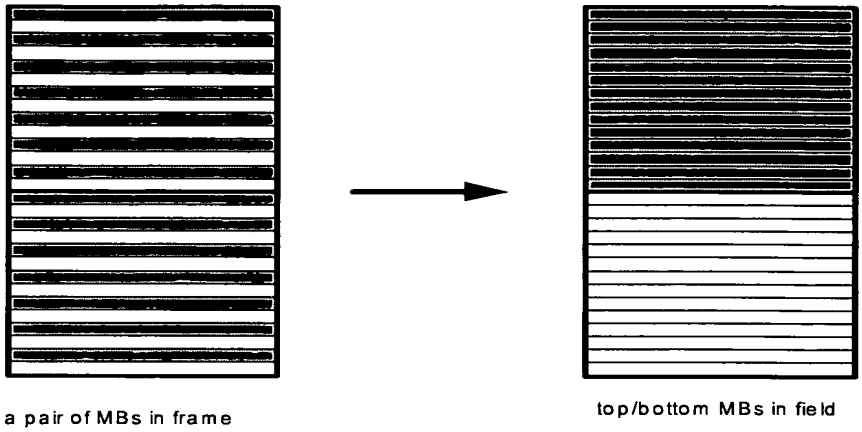


Figure 8-3 – Split of a pair of macroblocks into one top -field macroblock and one bottom -field macroblock. [Ed. Note: Should hae 16 lines per MB, not 14 as shown above]

When mb_field_decoding_flag is equal to 0, the top macroblock of a macroblock pair is decoded first, followed by the bottom macroblock. When mb_field_decoding_flag is equal to 1, the top-field macroblock is decoded first, followed by the bottom-field macroblock (see Figure 8-3).

When mb_frame_field_adaptive_flag is equal to 1, if only one macroblock of a macroblock pair is skipped, it shall be skipped in frame decoding mode if the other macroblock of the macroblock pair is coded in frame decoding mode otherwise it shall be skipped in field decoding mode. If both macroblocks in a macroblock pair are skipped, the process is described in subclause 8.2.8.1 for P and SP slices and in subclause 8.2.8.2 for B slices.

[Ed. Note: Need a precise description of macroblock numbering in MB -AFF case.]

8.2.8.1 Derivation process for frame or field decoding mode for a skipped macroblock pair in P or SP slices

[Ed.Note: this should go somewhere else in the spec ... to subclause 8.4]

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If both macroblocks in a macroblock pair are skipped in a P or SP slice, the macroblock pair will use the same frame or field decoding mode as its left neighbouring macroblock, if the left neighbouring macroblock exists and is in the same slice. If the left neighbouring macroblock is not available, the same frame or field decoding mode as the top neighbouring macroblock is used, if the top neighbouring macroblock exists and is in the same slice. Otherwise frame decoding mode is used.

8.2.8.2 Derivation process for frame or field decoding mode for skipped macroblock pair in B slice

[Ed.Note: this should go somewhere else in the spec ... to subclause 8.4]

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If both macroblocks in a macroblock pair are skipped in a B slice, th eir frame or field decoding mode is derived differently depending on the value of direct_spatial_mv_pred_flag. When direct_spatial_mv_pred_flag is equal to 1, the

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

frame or field decoding mode of the skipped macroblock pair is determined using the same meth od as for skipped macroblocks in P and SP slices, as described in subclause 8.2.8.1. When direct_spatial_mv_pred_flag is equal to 0, the frame or field decoding mode of the skipped macroblock pair shall follow the frame or field decoding mode of its co-located macroblock pair. The co-located macroblock is determined as described in subclause 8.4.1.4.2.1. [Ed. Note: Perhaps this should be moved later as co-located macroblock hasn't been defined yet.]

8.3 Intra prediction process

This process is invoked for I and SI macroblock types.

Input to this process are the macroblock type of the current macroblock (mb_type), prev_intra4x4_pred_mode_flag (if available), rem_intra4x4_pred_mode (if available), intra_chrom a_pred_mode (if available), and samples from neighbouring macroblocks or macroblock pairs.

Output of this process are Intra predicted samples of macroblocks or macroblock pairs.

The decoding process of Intra prediction modes is described. Samples used in the Intra prediction process shall be sample values prior to alteration by any deblocking filter operations.

8.3.1 Intra_4x4 prediction process

This process is invoked when mb_part_pred_mode(mb_type, 0) is equal to Intra_4x4. It is specified how the Intra predicted luma samples for the current macroblock are derived.

Input to this process are the syntax elements prev_intra4x4_pred_mode_flag, rem_intra4x4_pred_mode (if available), and decoded samples prior to the deblocking process from neighbouring luma blocks (if available).

Output of this process are Intra predicted luma samples for the current macroblock.

The luma component of a macroblock consists of 16 luma blocks of 4x4 samples. These luma blocks of 4x4 samples are scanned using the 4x4 luma block scanning process as specified in subclause 6.4.6.

For the each luma block of 4x4 samples, i.e., for 4x4LumaBlkIdx ranging from 0 to 15, the corresponding samples in adjacent blocks are derived in subclause 8.3.1.1 and the corresponding Intra4x4LumaPredMode[4x4LumaBlkIdx] is derived as specified in subclause 8.3.1.2. Then, given those values, the prediction values for each luma block are decoded as specified in subclause 8.3.1.3.

8.3.1.1 Derivation process for the available samples

8.3.1.1.1 Derivation process for the available samples for macroblocks

This process is invoked, when mb_adaptive_frame_field_flag is equal to 0. This process specifies the availabilty or non-availability and possible corresponding substitution for samples that are used for Intra prediction of a luma block of 4x4 samples.

Input to this process is the position of the luma block and prior (in decoding order) decoded samples prior to the deblocking filter process from adjacent luma blocks.

Output of this process are the values of samples labeled as A to M and their availability status.

Let 4x4LumaBlkLoc be the location of the upper left sample of the 4x4 luma block with index 4x4LumaBlkIdx.

The samples of a 4x4 luma block with index 4x4LumaBlkIdx containing samples a to p in Figure 8-4 are predicted using samples A to M in Figure 8-4 from adjacent blocks.

The sample a has the location 4x4LumaBlkLoc. T he sample d has the x-component of the location
4x4LumaBlkLoc[4x4LumaBlkIdx][0] + 4 and the y-component of the location
4x4LumaBlkLoc[4x4LumaBlkIdx][1].

Sample A is the sample with location 4x4LumaBlkLoc[4x4LumaBlkIdx][0] and
4x4LumaBlkLoc[4x4LumaBlkIdx][1] - 1.

[Ed.Note: Specify for samples B through M as for A or, better, change to description using P(x,y).]

[Ed.Note: not precise enough] Adjacent blocks containing samples A to D and I to M are the luma blocks to the left, above, and above-left to sample a. The adjacent block containing samples E to H is the luma block above -right to sample d.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

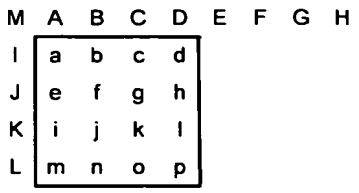


Figure 8-4 – Labelling of samples used for Intra_4x4 prediction

Samples A to M are marked as being "available for Intra prediction process" unless marked a "not available for prediction process" if one of the following conditions is true:

- if the sample is outside the picture or outside the current slice,
- if the sample is in a macroblock that is subsequent to the current macroblock in decoding order,
- if the sample is in a macroblock coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1.

When samples E-H are not available, and the sample D is marked as being ava ilable for prediction, the sample value of D is substituted for sample values E-H and samples E-H are marked as being available.

8.3.1.1.2 Derivation process for the available samples for macroblock pairs

This process is invoked, when mb_adaptive_frame_field_flag is equal to 1 and field_pic_flag is equal to 0.

For samples a to p in Figure 8-4, the samples A to M in Figure 8-4 are derived as follows.

When the current macroblock pair is coded frame decoding mode, the sa me process as described in subclause 8.3.1.1.1 is invoked.

When the current macroblock pair is coded field decoding mode, for the current top -macroblock, samples A to M are derived by:

Sample A is the sample with location 4x4Lum aBlkLoc[4x4LumaBlkIdx][0] and 4x4LumaBlkLoc[4x4LumaBlkIdx][1] - 2.

Sample I is the sample with location 4x4LumaBlkLoc[4x4LumaBlkIdx][0] - 1 and 4x4LumaBlkLoc[4x4LumaBlkIdx][1].

[Ed.Note: Specify for samples B through M as for A or, better, change to description using P(x,y).]

When the current macroblock pair is coded field decoding mode, for the current bottom -macroblock, samples A to M are derived by:

Sample A is the sample with location 4x4LumaBlkLoc[4x4LumaBlkIdx][0] and 4x4LumaBlkLoc [4x4LumaBlkIdx][1] - 1.

Sample I is the sample with location 4x4LumaBlkLoc[4x4LumaBlkIdx][0] - 1 and 4x4LumaBlkLoc[4x4LumaBlkIdx][1].

NOTE – Each 4x4 luma block is assumed to be reconstructed as a frame prior to decoding the next 4x4 luma block.

8.3.1.2 Derivation process for the Intra4x4LumaPredMode

Input to this process are the syntax elements prev_intra4x4_pred_mode_flag, rem_intra4x4_pred_mode, and the location 4x4LumaBlkLoc[4x4LumaBlkIdx] of a 4x4 luma block with 4x4LumaBlkIdx.

Output of this pro cess is the variable Intra4x4LumaPredMode[4x4LumaBlkIdx].

Table 8-7 specifies the values for Intra4x4LumaPredMode[4x4LumaBlkIdx] and the associated names.

Table 8-7 – Specification of Intra4x4LumaPredMode[4x4LumaBlkIdx] and associated names

Intra4x4LumaPredMode[4x4LumaBlkIdx]	Name of Intra4x4LumaPredMode[4x4LumaBlkIdx]

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

0	Intra_4x4_Vertical (prediction mode)
1	Intra_4x4_Horizontal (prediction mode)
2	Intra_4x4_DC (prediction mode)
3	Intra_4x4_Diagonal_Down_Left (prediction mode)
4	Intra_4x4_Diagonal_Down_Right (prediction mode)
5	Intra_4x4_Vertical_Right (prediction mode)
6	Intra_4x4_Horizontal_Down (prediction mode)
7	Intra_4x4_Vertical_Left (prediction mode)
8	Intra_4x4_Horizontal_Up (prediction mode)

Intra4x4LumaPredMode[4x4LumaBlkIdx] labelled 0, 1, 3, 4, 5, 6, 7, and 8 represent directions of predictions as illustrated in Figure 8-5.

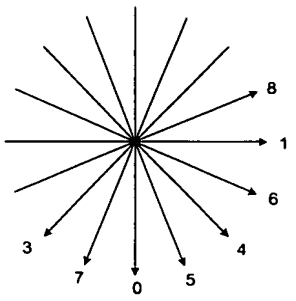


Figure 8-5 – Intra_4x4 prediction mode directions (informative)

The derivation process for the Intra4x4LumaPredMode for the current macroblock (pair) uses 3 steps:

Step 1: Derivation process for neighbouring macroblock (pair) indices and their availability.

Step 2: Derivation process for the Intra_4x4 prediction modes from neighboring macroblocks (pairs)

Step 3: Derivation process for Intra4x4LumaPredMode for the current macroblock (pair)

8.3.1.2.1 Derivation process for neighbouring macroblock addresses and their availability

This process is invoked, when mb_adaptive_frame_field is equal to 0 or field_pic_flag is equal to 1.

The input to this process is the address of the current macroblock (MbAddress).

The outputs of this process are the address of the macroblock to the left (Mb AddressA) of the current macroblock and the address of the macrolock above (MbAddressB) the current macroblock and their availability status.

The derivation process of the availability for neighbouring macroblocks in subclause 6.4.7 is invoked twice.

Input to the process in subclause 6.4.7 is MbAddressA = MbAddress – 1 and the output is whether the macroblock MbAddressA is available.

NOTE - MbAddressA is the address of the macroblock to the left of the current macroblock if it is available.

Input to the process in subclause 6.4.7 is MbAddressB = MbAddress – (frame_width_in_mbs_minus1 + 1) and the output is whether the macroblock MbAddressB is available.

NOTE - MbAddressB is the address of the macroblock above the current macroblock if it is available.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

8.3.1.2.2 Derivation process for neighbouring macroblock pair indices and their availability

This process is invoked, when mb_adaptive_frame_field is equal to 1 and field_pic_flag is equ al to 0.

The input to this process is the address of the current macroblock pair (MbPairAddress).

The outputs of this process are the address of the macroblock pair to the left (MbPairAddressA) of the current macroblock pair and the address of the macrolo ck above (MbPairAddressB) the current macroblock pair and their availability status.

The derivation process of the availability for neighbouring macroblock pairs in subclause 6.4.7 is invoked twice.

Input to the process in subclause 6.4.7 is MbPairAddressA = MbPairAddress – 1 and the output is whether the macroblock pair MbPairAddressA is available.

NOTE - MbPairAddressA is the address of the macroblock pair to the left of the current macroblock pair if it is available.

Input to the process in subclause 6.4.7 is MbPairAddressB = MbPairAddress – (frame_width_in_mbs_minus1 + 1) and the output is whether the macroblock pair MbPairAddressB is available.

NOTE - MbPairAddressB is the address of the macroblock pair above the current macroblock pair if it is available.

8.3.1.2.3 Derivation process for the Intra_4x4 prediction modes from neighboring macroblocks

This process is invoked, when mb_adaptive_frame_field is equal to 0 or field_pic_flag is equal to 1.

8.3.1.2.3.1 Derivation process for the Intra_4x4 prediction modes from left neighbouring macroblock

Input to this process is MbAddressA.

Output of this process are 4 values of the variable ExtIntra4x4LumaPredMode.

NOTE - ExtIntra4x4LumaPredMode is a 5x5 array that is used for the prediction of Intra4x4LumaPredMode, which is a 4x4 array. This process specifies 4 entries of the left column of ExtIntra4x4LumaPredMode.

When the macroblock MbAddressA is available and the mb_type of the macroblock MbAddress A is equal to I_4x4 or SI, the variable ExtIntra4x4LumaPredMode is derived in a process equivalent to:

```
for( i = 0, i < 4, i++ )
    ExtIntra4x4LumaPredMode[ (i+1)*5 ] = Intra4x4LumaPredMode[ 4x4LumaBlkScan( 3, i ) ]
```

where Intra4x4LumaPredMod is the variable assigned to the macroblock with index MbAddressA.

When macroblock MbAddressA is not available or the mb_type of the macroblock with index MbAddressA is not equal to I_4x4 or SI, the variable ExtIntra4x4LumaPredMode is derived in a process equivalent to:

```
for( i = 0, i < 4, i++ )
    ExtIntra4x4LumaPredMode[ (i+1)*5 ] = 2
```

8.3.1.2.3.2 Derivation process for the Intra_4x4 prediction modes from above neighbouring macroblock

Input to this process is MbAddressB.

Output of this process are 4 values of the variable ExtIntra4x4L umaPredMode.

NOTE - ExtIntra4x4LumaPredMode is a 5x5 array that is used for the prediction of Intra4x4LumaPredMode, which is a 4x4 array. This process specifies 4 entries of the top row of ExtIntra4x4LumaPredMode.

When the macroblock MbAddressB is availabl e and the mb_type of the macroblock MbAddressB is equal to I_4x4 or SI, the variable ExtIntra4x4LumaPredMode is derived in a process equivalent to:

```
for( i = 0, i < 4, i++ )
    ExtIntra4x4LumaPredMode[ i+1 ] = Intra4x4LumaPredMode[ 4x4LumaBlkScan( i, 3 ) ]
```

where Intra4x4LumaPredMode is the variable assigned to the macroblock MbAddressB.

When macroblock MbAddressA is not available or the mb_type of the macroblock MbAddressB is not equal to I_4x4 or SI, the variable ExtIntra4x4LumaPredMode is derived in a proc ess equivalent to:

```
for( i = 0, i < 4, i++ )
    ExtIntra4x4LumaPredMode[ i+1 ] = 2
```

8.3.1.2.4 Derivation process for the Intra_4x4 prediction modes from neighboring macroblock pairs

This process is invoked, when mb_adaptive_frame_field is equal to 1 and field_pic_flag is equal to 0.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

8.3.1.2.4.1 Derivation process for the Intra_4x4 prediction modes from neighboring macroblock pairs when current macroblock pair is coded in frame decoding mode

This process is invoked, when the current macroblock pair is coded in frame decoding mode.

Input to this process are MbPairAddress, MbPairAddressA, and MbPairAddressB.

Output of this process are 8 values of the variable ExtIntra4x4LumaPredModeTop for the top macroblock of the current macroblock pair and ExtIntra4x4LumaPredModeBottom for the bot tom macroblock of the current macroblock pair

The derivation processes as specified in subclause 8.3.1.2.3 are both invoked for both macroblocks of the current macroblock pair indexed as MbPairAddress.

For the derivation process in subclause 8.3.1.2.3.1, for the top macroblock of the current macroblock pair MbPairAddress, the top macroblock of the macroblock pair MbPairAddressA is assigned to MbAddressA as the input. The ouput is assigned to ExtIntra4x 4LumaPredModeTop.

For the derivation process in subclause 8.3.1.2.3.2, for the top macroblock of the current macroblock pair MbPairAddress, the bottom macroblock of the macroblock pair MbPairAddressB is assigned to MbAddressB as the input. The ouput is assigned to ExtIntra4x4LumaPredModeTop.

For the derivation process in subclause 8.3.1.2.3.1, for the bottom macroblock of the current macroblock pair MbPairAddress, the bo ttom macroblock of the macroblock pair MbPairAddressA is assigned to MbAddressA as the input. The ouput is assigned to ExtIntra4x4LumaPredModeBottom.

For the derivation process in subclause 8.3.1.2.3.1, for the bottom macroblock of the current macroblock pair MbPairAddress, the top macroblock of the current macroblock pair MbPairAddress is assigned to MbAddressB as the input. The ouput is assigned to ExtIntra4x4LumaPredModeBottom.

8.3.1.2.4.2 Derivation process for the Intra_4x4 prediction modes from neighboring macroblock pairs when current macroblock pair is coded in field decoding mode

This process is invoked, when the current macroblock pair is coded in frame decoding mode.

Input to this process are MbPairAddress, MbPairA ddressA, and MbPairAddressB.

Output of this process are 8 values of the variable ExtIntra4x4LumaPredModeTop for the top macroblock of the current macroblock pair and ExtIntra4x4LumaPredModeBottom for the bottom macroblock of the current macroblock pair.

The derivation processes as specified in subclause 8.3.1.2.3 are both invoked for both macroblocks of the current macroblock pair indexed as MbPairAddress.

For the derivation process in subclause 8.3.1.2.3.1, for the top macroblock of the current macroblock pair MbPairAddress, the top macroblock of the macroblock pair MbPairAddressA is assigned to MbAddressA as the input. The ouput is assigned to ExtIntra4x4LumaPredModeTop.

For the derivation process in subclause 8.3.1.2.3.1, for the bottom macroblock of the current macroblock pair MbPairAddress, the bottom macroblock of the macroblock pair MbPairAddressA is assigned to MbAddressA as the input. The ouput is assigned to ExtIntra4x4LumaPredModeBottom.

When the macroblock pair MbPairAddressB is coded in field decoding mode or when the macroblock with MbPairAddressB is not available, the following applies.

For the derivation process in subclause 8.3.1.2.3.2, for the top macroblock of the current macroblock pair MbPairAddress, the top macroblock of the macroblock pair MbPairAddressB is assigned to MbAddressB as the input. The ouput is assigned to ExtIntra4x4Lu maPredModeTop.

For the derivation process in subclause 8.3.1.2.3.2, for the bottom macroblock of the current macroblock pair MbPairAddress, the bottom macroblock of the macroblock pair MbPairAddressB is assigned to MbAddressB as the input. The ouput is assigned to ExtIntra4x4LumaPredModeBottom.

When the macroblock pair MbPairAddressB is coded in frame decoding mode, the following applies.

For the derivation process in subclause 8.3.1.2.3.2, for the top macroblock of the current macroblock pair MbPairAddress, the bottom macroblock of the macroblock pair MbPairAddressB is assigned to MbAddressB as the input. The ouput is assigned to ExtIntra4x4LumaPredModeTop.

For the derivation process in subclause 8.3.1.2.3.2, for the bottom macroblock of the current macroblock pair MbPairAddress, the bottom macroblock of the macroblock pair MbPairAddressB is assigned to MbAddressB as the input. The ouput is assigned to ExtIntra4x4LumaPredModeBottom.

Figure 8-6 illustrates the avove process when the current macroblock pair is coded in field decoding mode.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

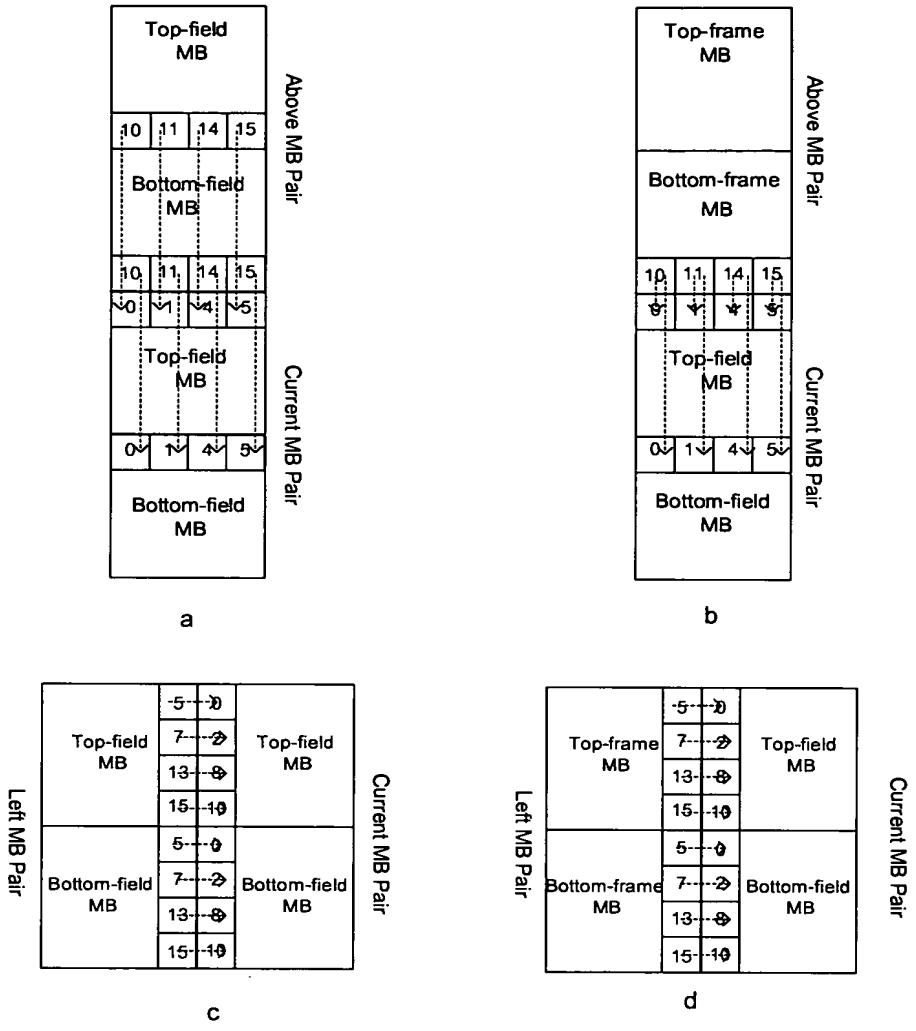


Figure 8-6 a) Definition of above neighbouring blocks when the above macroblock pair is in field mode b) Definition of above neighbouring blocks when the above macroblock pair is in frame mode c) Definition of left neighbouring blocks when the left macroblock pair is in field mode d) Definition of left neighbouring blocks when the left macroblock pair is in frame mode (informative)

8.3.1.2.5 Derivation process for Intra4x4LumaPredMode for the current macroblock

This process is invoked, when mb_adaptive _frame_field is equal to 0 or field_pic_flag is equal to 1.

The input to this process are prev_intra4x4_pred_mode_flag, rem_intra4x4_pred_mode, and ExtIntra4x4LumaPredMode.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

The variable Intra4x4LumaPredMode which is assigned to the current macroblock with index MbAddress is derived as follows.

```
for( 4x4LumaBlkIdx = 0, 4x4LumaBlkIdx < 16, 4x4LumaBlkIdx++ ) {
    i = 4x4LumaBlkIdx + RasterTo4x4LumaBlkOffset( 4x4LumaBlkIdx )
    Ext4x4LumaBlkIdx = 5 * ( i / 4 + 1 ) + ( i % 4 ) + 1
    PredIntra4x4LumaPredMode = Min( ExtIntra4x4LumaPredMode[ Ext4x4LumaBlkIdx - 1 ],
                                    ExtIntra4x4LumaPredMode[ Ext4x4LumaBlkIdx - 5 ] )
    if( prev_intra4x4_pred_mode_flag[ 4x4LumaBlkIdx ] )
        Intra4x4LumaPredMode[ 4x4LumaBlkIdx ] = PredIntra4x4LumaPredMode
    else {
        if( rem_intra4x4_pred_mode[ 4x4LumaBlkIdx ] < PredIntra4x4LumaPredMode )
            Intra4x4LumaPredMode[ 4x4LumaBlkIdx ] = rem_intra4x4_pred_mode[ 4x4LumaBlkIdx ]
        else
            Intra4x4LumaPredMode[ 4x4LumaBlkIdx ] = rem_intra4x4_pred_mode[ 4x4LumaBlkIdx ] + 1
    }
    ExtIntra4x4LumaPredMode[ Ext4x4LumaBlkIdx ] = Intra4x4LumaPredMode[ 4x4LumaBlkIdx ]
}
```

8.3.1.2.6 Derivation process for Intra4x4LumaPredMode for the current macroblock pair

This process is invoked, when mb_adaptive_frame_field is equal to 1 and field_pic_fla g is equal to 0.

The variable prev_intra4x4_pred_mode_flag, rem_intra4x4_pred_mode, and ExtIntra4x4LumaPredModeTop which are assigned to the top macroblock of the current macroblock pair are assigned as the input to the derivation process in subclause 8.3.1.2.5 with the output being Intra4x4LumaPredMode assigned to the top macroblock of the current macroblock pair.

The variable prev_intra4x4_pred_mode_flag, rem_intra4x4_pred_mode, and ExtIntra4x4LumaPredModeBottom which are assigned to the bottom macroblock of the current macroblock pair are assigned as the input to the derivation process in subclause 8.3.1.2.5 with the output being Intra4x4LumaPredMode assigned to the bottom macroblock of the current macroblock pair.

8.3.1.3 Decoding process for Intra_4x4 prediction modes

Input to this process are the samples A to M as specified in subclause 8.3.1.1 and Intra4x4LumaPredMode[4x4LumaBlkIdx] as specified in subcaluse 8.3.1.2.

Output of this process are the predicted samples for the 4x4 luma block with index 4x4LumaBlkIdx of the current macroblock.

Depending on Intra4x4LumaPredMode[4x4LumaBlkIdx], one of the Intra_4x4 prediction modes specified in subclauses 8.3.1.3.1 to 8.3.1.3.9 shall be used.

This process shall be invoked for each of the 4x4 luma blocks followed by the transform decoding process and picture construction process prior to deblocking f or each 4x4 luma block.

8.3.1.3.1 Specification of Intra_4x4_Vertical prediction mode

This Intra_4x4 prediction mode shall be used, when Intra4x4LumaPredMode[4x4LumaBlkIdx] is equal to 0.

This mode shall be used only if A, B, C, D are available. The values of the predicted samples a to p are derived as follows:

- a, e, i, m are equal to A,
- b, f, j, n are equal to B,
- c, g, k, o are equal to C,
- d, h, l, p are equal to D.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

8.3.1.3.2 Specification of Intra_4x4_Horizontal prediction mode

This Intra_4x4 prediction mode sh all be used, when Intra4x4LumaPredMode[4x4LumaBlkIdx] is equal to 1.

This mode shall be used only if I, J, K, L are available. The values of the predicted samples a to p are derived as follows:

- a, b, c, d are equal to I,
- e, f, g, h are equal to J,
- i, j, k, l are equal to K,
- m, n, o, p are equal to L.

8.3.1.3.3 Specification of Intra_4x4_DC prediction mode

This Intra_4x4 prediction mode shall be used, when Intra4x4LumaPredMode[4x4LumaBlkIdx] is equal to 2.

The values of the predicted samples a to p are derived as follows:

If all samples A, B, C, D, I, J, K, L are available, all predicted samples are equal to (A+B+C+D+I+J+K+L+4)>>3.

If A, B, C, and D are not available and I, J, K, and L are available, all predicted samples are equal to (I+J+K+L+2)>>2.

If I, J, K, and L are not available and A, B, C, and D are available, all predicted samples are equal to (A+B+C+D+2)>>2.

If all samples A, B, C, D, I, J, K, L are not available, all predicted samples are equal to 128.

A block may therefore always be pre dicted in this mode.

8.3.1.3.4 Specification of Intra_4x4_Diagonal_Down_Left prediction mode

This Intra_4x4 prediction mode shall be used, when Intra4x4LumaPredMode[4x4LumaBlkIdx] is equal to 3.

This mode shall be used only if all A, B, C, D, I, J, K, L, M are ava ilable. The values of the predicted samples a to p are derived as follows:

- a is equal to (A + 2B + C + 2) >> 2
- b, e are equal to (B + 2C + D + 2) >> 2
- c, f, i are equal to (C + 2D + E + 2) >> 2
- d, g, j, m are equal to (D + 2E + F + 2) >> 2
- h, k, n are equal to (E + 2F + G + 2) >> 2
- l, o are equal to (F + 2G + H + 2) >> 2
- p is equal to (G + 3H + 2) >> 2

8.3.1.3.5 Specification of Intra_4x4_Diagonal_Down_Right prediction mode

This Intra_4x4 prediction mode shall be used, when Intra4x4LumaPred Mode[4x4LumaBlkIdx] is equal to 4.

This mode shall be used only if all A, B, C, D, I, J, K, L, M are available. The values of the predicted samples a to p are derived as follows:

- m is equal to (J + 2K + L + 2) >> 2
- i, n equal to (I + 2J + K + 2) >> 2
- e, j, o are equal to (M + 2I + J + 2) >> 2
- a, f, k, p are equal to (A + 2M + I + 2) >> 2
- b, g, l are equal to (M + 2A + B + 2) >> 2
- c, h are equal to (A + 2B + C + 2) >> 2
- d is equal to (B + 2C + D + 2) >> 2

8.3.1.3.6 Specification of Intra_4x4_V ertical_Right prediction mode

This Intra_4x4 prediction mode shall be used, when Intra4x4LumaPredMode[4x4LumaBlkIdx] is equal to 5.

This mode shall be used only if all A, B, C, D, I, J, K, L, M are inside the slice. The values of the predicted samples a to p are derived as follows:

- a, j are equal to (M + A + 1) >> 1
- b, k are equal to (A + B + 1) >> 1

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

- c, l are equal to (B + C + 1) >> 1
- d is are equal to (C + D + 1) >> 1
- e, n are equal to (I + 2M + A + 2) >> 2
- f, o are equal to (M + 2A + B + 2) >> 2
- g, p are equal to (A + 2B + C + 2) >> 2
- h is equal to (B + 2C + D + 2) >> 2
- i is equal to (M + 2I + J + 2) >> 2
- m is equal to (I + 2J + K + 2) >> 2

8.3.1.3.7 Specification of Intra_4x4_Horizontal_Down prediction mode

This Intra_4x4 prediction mode shall be used, when Intra4x4LumaPredMode[4x4LumaBlkIdx] is equal to 6.

This mode shall be used only if all A, B, C, D, I, J, K, L, M are available. The values of the predicted samples a to p are derived as follows:

- a, g are equal to (M + I + 1) >> 1
- b, h are equal to (I + 2M + A+ 2) >> 2
- c is equal to (M + 2A + B+ 2) >> 2
- d is equal to (A + 2B + C+ 2) >> 2
- e, k are equal to (I + J + 1) >> 1
- f, l are equal to (M + 2I + J+ 2) >> 2
- i, o are equal to (J + K + 1) >> 1
- j, p are equal to (I + 2J + K+ 2) >> 2
- m is equal to (K + L + 1) >> 1
- n is equal to (J + 2K + L + 2) >> 2

8.3.1.3.8 Specification of Intra_4x4_Vertical_Left prediction mode

This Intra_4x4 prediction mode shall be used, when Intra4x4LumaPredMode[4x4LumaBlkIdx] is equal to 7.

This mode shall be used only if all A, B, C, D, I, J, K, L, M are available. The values of the predicted samples a to p are derived as follows:

- a is equal to (A + B + 1) >> 1
- b, i are equal to (B + C + 1) >> 1
- c, j are equal to (C + D + 1) >> 1
- d, k are equal to (D + E + 1) >> 1
- l is equal to (E + F + 1) >> 1
- e is equal to (A + 2B + C + 2) >> 2
- f, m are equal to (B + 2C + D + 2) >> 2
- g, n are equal to (C + 2D + E + 2) >> 2
- h, o are equal to (D + 2E + F + 2) >> 2
- p is equal to (E + 2F + G + 2) >> 2

8.3.1.3.9 Specification of Intra_4x4_Horizontal_Up prediction mode

This Intra_4x4 prediction mode shall be used, when Intra4x4LumaPredMode[4x4LumaBlkIdx] is equal to 8.

This mode shall be used only if all A, B, C, D, I, J, K, L, M are available. The values of the predicted samples a to p are derived as follows:

- a is equal to (I + J + 1) >> 1
- b is equal to (I + 2J + K + 2) >> 2
- c, e are equal to (J + K + 1) >> 1
- d, f are equal to (J + 2K + L + 2) >> 2
- g, i are equal to (K + L + 1) >> 1

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

- h, j are equal to (K + 3L + 2) >> 2
- k, l, m, n, o, p are equal to L

8.3.2 Intra_16x16 prediction process

This process is invoked when mb_part_pred_mode(mb_type, 0) is equal to Intra_16x16. It is specified how the Intra predicted luma samples for the current macroblock are derived.

Input to this process are the syntax elements mb_type, and decoded samples prior to the deblocking process from neighbouring luma blocks (if available).

Output of this process are Intra predicted luma samples for the current macroblock.

[Ed.Note: what about MB-AFF ???]

Let P(-1,y), with y=-1,0,...,15 and P(x,-1), with x=0,...,15 denote the 33 neighbouring samples immediately above and/or to the left of the 16x16 luma block. Let Pred(x,y) x,y = 0..15 denote the prediction for the 16x16 luma block samples. There are 4 different prediction modes as specified in subclauses **Error! Reference source not found.** to **Error! Reference source not found.**. Only Mode 3 uses P(-1,-1).

Samples P(x,-1) or P(-1,y) shall be considered not available if they are outside the picture or outside the current slice, or if they are in a non-intra macroblock and constrained_intra_pred is 1. If none of the neighbouring samples are available, Mode 2 (DC prediction) is the only possible 16x16 luma prediction mode.

8.3.2.1 Mode 0: vertical prediction process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

This mode shall be used only if all neighbouring samples P(x, -1) , with x = 0..15 are available.

$$Pred(x, y) = P(x, -1), \text{ with } x, y = 0..15 \tag{8-10}$$

8.3.2.2 Mode 1: horizontal prediction process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

This mode shall be used only if all neighbouring samples P(-1, y) , y=0..15 are available.

$$Pred(x, y) = P(-1, y), \text{ with } x, y = 0..15 \tag{8-11}$$

8.3.2.3 Mode 2: DC prediction process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If all neighbouring samples P(x', -1) and P(-1, y') used in Equation 8-12 are available, the prediction for all luma samples in the macroblock is given by:

$$Pred(x, y) = [\sum_{x'=0}^{15} P(x',-1) + \sum_{y'=0}^{15} P(-1, y') + 16] >> 5 \quad \text{with } x, y = 0..15 \tag{8-12}$$

If the neighbouring samples P(x', -1) are not available and the neighbouring samples P(-1, y') are available, the prediction for all luma samples in the macroblock is given by:

$$Pred(x, y) = [\sum_{y'=0}^{15} P(-1, y') + 8] >> 4 \quad \text{with } x, y = 0..15, \tag{8-13}$$

If the neighbouring samples P(-1, y') are not available and the neighbouring samples P(x', -1) are available, the prediction for all luma samples in the macroblock is given by:

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

$$\text{Pred}(x, y) = [\sum_{x'=0}^{15} P(x',-1) + 8] >> 4 \quad \text{with } x, y = 0..15, \tag{8-14}$$

If none of the neighbouring samples P(x', -1) and P(-1,y') are available, the prediction for all luma samples in the macroblock shall be 128.

8.3.2.4 Mode 3: plane prediction process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

This mode shall be used only if all neighbouring samples P(x, -1) with x = -1..15 and P(-1, y) with y = 0..15 are available:

$$\text{Pred}(x,y) = \text{Clip1}((a + b (x-7) + c (y-7) +16) >> 5), \tag{8-15}$$

where:

$$a = 16 (P(-1,15) + P(15,-1)) \tag{8-16}$$

$$b = (5*H+32)>>6 \tag{8-17}$$

$$c = (5*V+32)>>6 \tag{8-18}$$

and H and V are specified in Equations 8-19 and 8-20.

$$H = \sum_{x=1}^{8} x \cdot (P(7 + x,-1) - P(7 - x,-1)) \tag{8-19}$$

$$V = \sum_{y=1}^{8} y \cdot (P(-1,7 + y) - P(-1,7 - y)) \tag{8-20}$$

8.3.3 Intra prediction process for chroma blocks

This process is invoked for I and SI macroblock types. It is specified how the Intra predicted chroma samples for the current macroblock are derived.

Input to this process are the syntax elements intra_chroma_pred_mode and decoded samples prior to the deblocking process from neighboring luma blocks (if available).

Output of this process are Intra predicted chroma samples for the current macroblock.

[Ed.Note: what about MB-AFF ???]

The chroma in intra macroblocks is predicted in a manner very similar to the luma block in Intra_16x16 macroblock type (subclause **Error! Reference source not found.**), using one of four prediction modes. The same prediction mode is applied to both chroma blocks, but it is independent of the prediction mode used for the luma.

NOTE - If any portion of the luma macroblock is coded in intra mode, the entire chroma macroblock is coded intra.

Let P(-1,y), y=-1,0,...,7 and P(x,-1), with x=-1,0,...,7 denote the 17 neighbouring samples immediately above and/or to the left of the 8x8 chroma block. Let Pred(x,y) with x,y = 0..7 denote the prediction for the 8x8 chroma block samples. There are four different prediction modes as specified in subclauses **Error! Reference source not found.** to **Error! Reference source not found.**. Only Mode 3 uses P(-1,-1).

Samples P(x,-1) or P(-1,y) shall be considered not available if they are outside the picture or outside the current slice, or if they are in a non-intra macroblock and constrained_intra_pred is 1. If none of the neighbouring samples are available, Mode 2 (DC prediction) is the only possible 8x8 chroma prediction mode.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

8.3.3.1 Mode 0: DC prediction process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If all samples P(-1,*n*) and P(*n*,-1) used in Equation 8-21 are available, the prediction is formed as:

$$\text{Pred}(x, y) = \left(\left(\sum_{n=0}^{7}\left(P(-1,n) + P(n,-1)\right)\right) + 8\right) >> 4 \quad \text{with } x, y = 0..7, \tag{8-21}$$

If the 8 samples P(-1,*n*) are not available but the 8 samples P(*n*,-1) are available, the prediction is formed as:

$$\text{Pred}(x,y) = \left[\left(\sum_{n=0}^{7} P(n,-1)\right) + 4\right] >> 3 \text{ with } x, y = 0..7, \tag{8-22}$$

If the 8 samples P(*n*,-1) are not available but the 8 samples P(-1, *n*) are available, the prediction is formed as:

$$\text{Pred}(x,y) = \left[\left(\sum_{n=0}^{7} P(-1,n)\right) + 4\right] >> 3 \text{ with } x, y = 0..7, \tag{8-23}$$

If none of the neighbouring samples P(x, -1) and P(-1,y) are available, the prediction for all chroma samples in the macroblock shall be 128.

8.3.3.2 Mode 1: horizontal prediction process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

This mode shall be used only if all neighbouring samples P(-1, y), with y = 0..7 are available:

$$\text{Pred}(x, y) = P(-1, y), \text{ with } x, y = 0..7 \tag{8-24}$$

8.3.3.3 Mode 2: vertical prediction process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

This mode shall be used only if all neighbouring samples P(x, -1), with x = 0..7 are available:

$$\text{Pred}(x, y) = P(x, -1), \text{ with } x, y = 0..7 \tag{8-25}$$

8.3.3.4 Mode 3: plane prediction

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

For the plane mode, the prediction is formed as:

$$\text{Pred}(x,y) = \text{Clip1}((a + b (x-3) + c (y-3) + 16) >> 5), \text{ with } x, y = 0,\ldots,7 \tag{8-26}$$

where:

$$a = 16 (P(-1,7) + P(7,-1)) \tag{8-27}$$

$$b = (17*H+16)>>5 \tag{8-28}$$

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

$$c = (17*V+16)>>5 \tag{8-29}$$

and H and V are specified as:

$$H = \sum_{x=1}^{4} x \cdot [P(3+x,-1) - P(3-x,-1)] \tag{8-30}$$

$$V = \sum_{y=1}^{4} y \cdot [P(-1,3+y) - P(-1,3-y)] \tag{8-31}$$

8.4 Inter prediction process

This process is invoked, when decoding P, SP, or B slices, and a macroblock is skipped or the macroblock type indicates Pred_L0, Pred_L1, Direct or BiPred prediction modes.

Input to this process are the skipped ma croblock indication (if available), the macroblock type mb_type (if available), the motion vector differences mvd_l0 and mvd_l1 (if available), the reference index list 0 and the reference index list 1 from subclause 8.2.6 and the reference pictures in the decoded picture buffer, the reference indices ref_idx_l0 and ref_idx_l1 (if available).

Output of this process are Inter prediction samples.

The Inter prediction process specifies a prediction of samples using reference picture s in the decoded picture buffer and syntax elements.

The inter prediction process for a macroblock or its partition consists of

- Calculation of variables for luma and chroma motion vector components and reference picture indices as specified in subclause 8.4.1.
- Calculation of luma and chroma inter prediction samples as specified in subclause 8.4.2.

In P and SP slices, inter prediction is only list 0 prediction. In B slices, inter prediction may be list 0 prediction only, list 1 prediction only, or bi -prediction using both list 0 and list 1 prediction.

8.4.1 Derivation process for motion vector components and reference picture indices

[Ed. Note: Perhaps this should be divided into separate sections for P and B slices. And perhaps all of the detail doesn't belong here, but in other sections.]

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

The inter prediction process for each macroblock is specified by the syntax elements mb_type and, if present, sub_mb_type [i], with i indicating the sub -macroblock index.

Further, the inter prediction process for each macroblock is specified by several syntax elements, refere nce picture index for list 0 prediction ref_idx_l0 [i] and reference picture index for list 1 prediction ref_idx_l1 [i], with the macroblock type being given by the slice_type dependent value of mb_type and i being specified by num_mb_part(mb_type) in subclause 7.4.5.1.

Further, the inter prediction process for each macroblock is specified by the syntax elements motion vector difference for list 0 prediction mvd_l0[i][j][k] and motion vector difference for list 1 predic tion mvd_l1[i][j][k] with i being specified by num_mb_part(mb_type) in subclause 7.4.5.1, j being the sub -macroblock index in the 8x8 macroblock partition and is equal to 0 if the 8x8 partition is not sub -divided, and k equals to 0 corresponds to the x -component of the motion vector difference and k equals to 1 corresponds to the y -component of the motion vector difference.

MvL0[i][j][k] and MvL1[i][j][k] are the motion vectors used for inter prediction of the macroblock, for the list 0 and list 1 predictors, respectively. MvpL0[i][j][k] and MvpL1[i][j][k] are the motion vector predictors formed for the macroblock, for the list 0 and list 1 predictors, respectively. If list 0 prediction is not use d for the macroblock partition, MvL0[i][j][k] is set to zero. Otherwise, MvL0[i][j][k] is calculated by adding the motion vector predictor for the macroblock partition or sub -macroblock partition to mvd_l0[i][j][k]. For B slices, if list 1 prediction is not used for the macroblock partition, MvL1[i][j][k] is set to zero. Otherwise, MvL1[i][j][k] is calculated by

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

adding the motion vector predictor for the macroblock partition or sub -macroblock partition to mvd_l1[i][j][k].
MvL0[i][j][k] = MvpL0[i][j][k] + mvd_l0[i][j][k]

MvL1[i][j][k] = MvpL1[i][j][k] + mvd_l1[i][j][k]

When decoding a slice, it is assumed that for all neighbouring macroblocks, the syntax elements mb_type, RefIdxL0 [i], RefIdxL1[i], MvL0[i][j][k], and MvL1[i][j][k] ref_idx_l0[i], ref_idx_l1[i], mvd_l0[i][j][k], mvd_l1[i][j][k] are stored as necessary to calculate the luma motion vector prediction.

RefIdxL0[i] and RefIdxL1[i] are the reference picture indices used for inter prediction for the macroblock partition, for the list 0 and list 1 predictors, respectively. If list 0 prediction is not used for the macroblock partition , RefIdxL0[i] is set to zero. If the ref_idx_l0[i] field is present, RefIdxL0[i] is equal to ref_idx_l0[i]. If the ref_idx_l0[i] field is not present because num_ref_idx_l0_active_minus1 is equal to 0, RefIdxL0[i] is equal to 0. For skipped macroblocks in P or SP slices, RefIdxL0[i] is derived as described in subclause 8.4.1.4.2.2. In B slices, if list 1 prediction is not used for the macroblock partition, RefIdxL1[i] is set to zero. I f the ref_idx_l1[i] field is present, RefIdxL1[i] is equal to ref_idx_l1[i]. If the ref_idx_l1[i] field is not present because num_ref_idx_l1_active_minus1 is equal to 0, RefIdxL1[i] is equal to 0. For skipped macroblocks in B slices or direct macroblocks or macroblock partitions, RefIdxL0[i] RefIdxL1[i] are derived using one of two methods. If direct_ spatial_mv_pred_flag is equal to one, the motion vector prediction is formed using the spatial method, as defined in subclause 8.4.1.4.2.2. If direct_spatial_mv_pred_flag is equal to zero, the motion vector prediction is formed using the temporal method, as defined in subclause 8.4.1.4.2.3.

For decoding future B slices in decoding order, when the macroblock is part of a slice which is part of a picture "used for reference" in the decoded picture buffer, the syntax elements mb_type, ref_idx_l0 [i], ref_idx_l1[i], mvd_l0[i][j][k], and mvd_l1[i][j][k] are stored together with the decoded pictures. If list 0 prediction is used for the macroblock partition or sub -macroblock partition, MvL0[i][j][k] and RefIdxL0[i] are stored. If list 0 prediction is not used for the macroblock partition or sub -macroblock partition but list 1 prediction is used, MvL1[i][j][k] and RefIdxL1[i] are stored. If neither list 0 or list 1 prediction is used, i.e. for I and SI macroblock types, the value 0 is stored for both the motion vector and the reference index. [Ed. Note: Should this be described here, or in subclause 8.4.1.4.2.3 ?] When direct_temporal_constrained_flag is equal to 1, for each 8x8 sub-macroblock, only the motion vectors of the corner 4x4 blocks MvL0[i][j][k] and MvL1[i][j][k] need to be stored, as described in subclause 8.4.1.4.2.3.

This paragraph specifies how the variables RefIdxL0 [i], RefIdxL1[i], MvL0[i][j][k], and MvL1[i][j][k] are obtained from the syntax elements mb_type, ref_idx_l0 [i], ref_idx_l1[i], mvd_l0[i][j][k], and mvd_l1[i][j][k] in the current and in prior decoded macroblocks in decoding order.

The calculation of motion vector components is done separately for the list 0 and list 1 predictors, if used, and consists of the following steps

- Determination of the neighbouring luma blocks with associated luma motion vectors
- Determination of the values of the luma motion vectors in the neighbouring luma blocks dependent on ref_idx_l0 and ref_idx_l1, etc
- Determination of the luma motion vector prediction method depending on the macroblock partition
- Calculation of luma motion vector components from predictors and differential vectors
- Calculation of the chroma motion vector components

8.4.1.1 Derivation process for the neighbouring blocks for motion vector and reference index prediction

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

No motion vector component prediction takes place across macroblock boundaries of macroblocks that do not belong to the same slice. For the purpose of motion vector component prediction, macroblocks that do not belong to the same slice are treated as outside the picture.

The prediction of motion vector components, MvpL0[i][j][k] and MvpL0[i][j][k] , for a luma block E is calculated based on the motion vectors, MvL0[i][j][k] and MvL1[i][j][k], and reference picture indices, RefIdxL0[i] and RefIdxL0[i], of neighbouring blocks A, B, C, and D as shown in Figure 8-7.

A The block containing the sample to the left of the upper left sample in E

B The block containing the sample just above the upper left sample in E

C The block containing the sample above and to the right of the upper right sample in E

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

D The block containing the sample above and to the left of the upper left sample in E

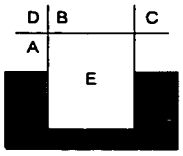


Figure 8-7 – Determination of the neighbouring luma blocks and associated motion vectors

The motion vector components of blocks A, B, C, and D are the input motion vector components for the prediction of the motion vector component of the current block E, unless they are modified by the rules in subclaus e 8.4.1.2.

[Ed. Note: Need a precise definition of the neighboring macroblocks in MB -AFF case. Some of the figures in VCEG-O37 might help.]

8.4.1.2 Derivation process for the luma motion vector predictors in the neighbouring blocks

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

8.4.1.2.1 Derivation process for the luma motion vector predictors in the neighbouring blocks in P and SP slices

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

The following rules determine the predicted motion vector value resulting from the median prediction process f or block E:

- If block C is outside the current picture or slice or is not available due to the decoding order within a macroblock as specified in **Error! Reference source not found.**, its motion vector and reference picture index shall be considered equal to the motion vector and reference picture index for block D.
- If blocks B, C, and D are all outside the current picture or slice, their motion vector values and reference picture indices shall be considered as equal to the motion v ector value and reference picture index for block A.
- If any predictor not specified by the first or second rules above is coded as intra or is outside the current picture or slice, its motion vector value shall be considered equal to zero and it shall be considered to have a different reference picture than block E.
- If only one of the three blocks A, B and C has the same reference picture as block E, then the predicted motion vector for block E shall be equal to the motion vector of the A, B, or C block with the same reference picture as block E; otherwise, each component of the predicted motion vector value for block E shall be the median of the corresponding motion vector component values for blocks A, B, and C.

When mb_adaptive_frame_field_flag is equ al to 1, the following additional rules apply:

- If any of the blocks A, B, C, and D (indicated as X in the following) is coded in field decoding mode, such block is assigned, in addition to its own field MV, two frame MVs for the purpose of motion vector prediction. The first assigned frame MV is the field MV of the block with vertical motion vector component multiplied by 2. The second assigned frame MV is the field MV of the block in the same geometric location as block X in the second macroblock of the m acroblock pair with vertical motion vector component multiplied by 2. If any of the blocks A, B, C, and D (indicated as X in the following) is coded in frame mode, such block is assigned, in addition to its own frame MV, two field MVs for the purpose of motion vector prediction. The first assigned field MV is the frame MV of the block with vertical motion vector component divided by 2. The second assigned field MV is the frame MV of the

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

block in same geometric location as block X in the second macroblock of the macroblock pair with vertical motion vector component divided by 2.

- If any of the blocks A, B, C, and D (indicated as X in the following) is coded in field mode, such block is assigned, in addition to its own field reference number, two reference fram e numbers for the purpose of motion vector prediction. The first assigned reference frame number is the reference field number of the block divided by 2. The second assigned reference frame number is the reference field number, divided by 2, of the block in same geometric location as block X in the second macroblock of the macroblock pair. If any of the blocks A, B, C, and D (indicated as X in the following) is decoded in frame mode, such block is assigned, in addition to its own frame reference number, two reference field numbers for the purpose of motion vector prediction. The first assigned reference field number is the reference frame number of the block multiplied by 2. The second assigned reference field number is the reference frame number, multiplied by 2, of the block in same geometric location as block X in the second macroblock of the macroblock pair.
- If block E is in frame mode and block A, B, C or D is also in frame mode, the frame MVs and the reference frame number of block A, B, C or D shall be used in calculating the motion vector prediction of block E. If block A, B, C, or D is in field mode and its two assigned reference frame numbers are the same, its two assigned frame MVs shall be averaged for use in calculating the motion vector predictio nof block E. In this case, the assigned reference frame number shall be used as the reference frame number of the field mode block for use in calculating the motion vector prediction of block E. If block A, B, C, or D is decoded in field mode and its two assigned reference frame numbers are not the same, the particular block A, B, C, or D shall be treated as being out of picture/slice for the purpose of predicting the motion vectors of block E.
- If block E is in field mode and block A, B, C or D is also in field mode, the field MVs and the reference field number of block A, B, C or D shall be used in calculating the motion vector prediction of block E. If block A, B, C, or D is in frame mode and its two assigned reference field numbers are the same, its two assigned field MVs shall be averaged for use in calculating the motion vector prediction of block E. In this case, the assigned reference field number shall be used as the reference field number of the frame coded block for use in calculating the motion ve ctor prediction of block E. If block A, B, C, or D is in frame mode and its two assigned reference field numbers are not the same, the particular block A, B, C, or D shall be treated as being out of picture/slice for the purpose of predicting the motion v ectors of block E.

8.4.1.2.2 Derivation process for the luma motion vector predictors in the neighbouring blocks in B slices

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

When decoding a B slice, the predictors of motion vector components for the list 0 and list 1 predictions are calculated as specified in subclause 8.4.1.2.1, with the following distinctions:

- the prediction for the motio n vector component in list i prediction which is referred to as MVC_i, where i = 0 or 1, is calculated only from neighbouring blocks A, B, C, and D that use reference pictures in the same temporal direction as the reference picture referred to by MV_i. If a neighbouring block does not have a motion vector referring to a reference picture in the same temporal direction, both components of the motion vector predictor for that block are set to zero and the neighbouring block shall be considered as belonging to a *different reference picture* for purposes of computing the median prediction of subclause 8.4.1.2.1.
- if a neighbouring block has two motion vectors both using the same reference picture as the reference picture associated with MV_i, then the list 0 motion vector from that block is used as a prediction. Otherwise, if a neighbouring block has both motion vectors using a reference picture in the same temporal direction as the reference picture associated with MV_i but they point to di fferent reference pictures, then the motion vector with the associated temporally closest reference picture is used.
- reconstructed motion vectors in direct mode neighbouring blocks, MvL0[i][j][k] and MvL1[i][j][k], shall be used as predictions for the current block motion vectors.
  - If a direct mode neighbouring block has two motion vectors, then this block is treated as if it were a bi - predictive neighbouring block.

If a direct mode neighbouring block has only one motion vector, then this blo ck is considered as a list 0 or list 1 block. (This occurs only when spatial direct mode is used.)

  - if both the list 0 and list 1 reference pictures are in the same temporal direction from the current picture and both are short term reference pictures, then the following process is followed for forming the predictor for the list 1 motion vector MV_{1_n} by scaling the list 0 motion vector, MV_0.

Formatted: Indent: Left: 0.25", Bulleted + Level: 1 + Aligned at: 0" + Tab after: 0.25" + Indent at: 0.25", Tabs: 0.5", List tab + Not at 0.25"

Formatted: Indent: Left: 0.25", Bulleted + Level: 1 + Aligned at: 0" + Tab after: 0.25" + Indent at: 0.25", Tabs: 0.5", List tab + Not at 0.25"

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

- The motion vector predictor for motion vector MV_1 is the scaled motion vector (smv), which is calculated from the motion vector MV_0 as:

  $Z = (TD_1 \times 256) / TD_0 \qquad smv = (Z \times MV_0 + 128) >> 8$

- where TD_0 is the DiffPicOrderCount from the current picture to the reference picture indicated by ref_idx_l0, i.e. DiffPicOrderCount(current_pic, ref_idx_l0), and TD_1 is the DiffPicOrderCount from the current picture to the reference picture indicated by ref_idx_l1, DiffPicOrderCount(current_pic, ref_idx_l1).

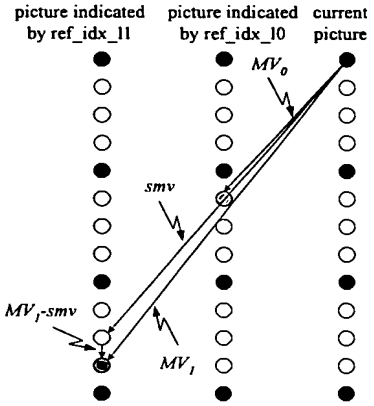


Figure 8-8 – Differential motion vector decoding with scaled motion vector

8.4.1.3 Derivation process for the luma motion vector prediction method

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

With exception of the 16x8 and 8x16 block shapes, "median prediction" (see subclause 8.4.1.3.1) is used. In case the macroblock may be classified to have directional segmentation the prediction is specified in subclause. The motion vector for a Skip mode macroblock shall be obtained as described in subclause 8.4.1.4.

8.4.1.3.1 Decoding process for median luma motion vector prediction

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

Specify median prediction.

The reference picture index is given by ref_idx_l0 and ref_idx_l1.

8.4.1.3.2 Decoding process for directional segmentation luma motion vector prediction

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

If the macroblock where the block to be predicted is coded in 16x8 or 8x16 mode, the prediction is generated as follows (refer to Figure 8-9 and the definitions of A, B, C, E above):

a) Motion vector block size 8x16:

   1) Left block: A is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

2) Right block: C is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used

b) Motion vector block size 16x8:

1) Upper block: B is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used

2) Lower block: A is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used

If the indicated prediction block is outside the picture, the same substitution rules are applied as in the case of me dian prediction.

For field-coded macroblocks, the directional segmentation follow the same conventions as the above, but the neighbouring blocks are constructed by using the rules mentioned in subclause 8.4.1.1??? [Ed. Note: ??].

The following additional considerations apply in the case of macroblock-adaptive frame/field coding When mb_field_decoding_flag == 1, block A, B, or C may be in different frame/field mode than block E. The reference picture of A, B or C used in determining if the individual block A, B or C is used as prediction or if median prediction is used is defined as follows.

1. Block E is in frame mode -- if any of the blocks A, B or C is in field mode and its two assigned reference frame numbers are the same, its two assigned frame motion vec tors are averaged for use in the motion vector prediction of block E. If the two assigned reference frame numbers are not the same, the block A, B or C is considered outside the picture.
2. Block E is in field mode -- if any of the blocks A, B or C is in fram e mode and its two assigned reference field numbers are the same, its two assigned field motion vectors are averaged for use in the motion vector prediction of block E. If the two assigned reference field numbers are not the same, the block A, B or C is co nsidered outside the picture.

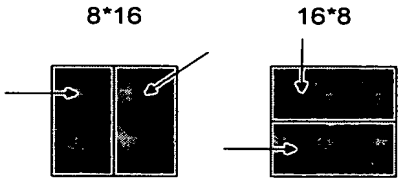


Figure 8-9 – Directional segmentation prediction

8.4.1.4 Derivation process for luma motion vectors for skipped and Direct prediction macroblocks

Input to this process is/are ??? [Ed.Note: speci fy input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

8.4.1.4.1 Derivation process for luma motion vectors for skipped macroblocks in P and SP slices

Input to this process is/are ??? [Ed.Note: specify input to this pro cess]

Output of this process is/are ??? [Ed.Note: specify output of this process]

When decoding a P or SP slice, the motion vector for a skipped macroblock shall be equal to the prediction of the motion vector that would be computed for this macroblock whe n decoding a macroblock coded as Pred_L0_16x16 macroblock type, MvL0[i][j][k] = MvpL0[i][j][k], except if any of the conditions below hold. In that case, a motion vector with both components being equal to 0, MvL0[i][j][k] = 0, shall be used instead:

a) The macroblock immediately above or to the left is not available (that is, is outside of the picture or belongs to a different slice)

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

b) Either one of the motion vectors applying to samples A or B (as described in subclause 8.4.1.1 uses picture with ref_idx_l0 == 0 as reference and has zero magnitude.

The reference picture index is given by ref_idx_l0 and ref_idx_l1.

8.4.1.4.2 Derivation process for luma motion vectors for skipped, Direct_16x16 and Direct_8x8 in B slices

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

When decoding a B slice, if the macroblock is skipped or coded as Direct_16x16 macroblock type or the sub -macroblock is coded as Direct_8x8 sub-macroblock type, the motion vectors for the corresponding partitions are derived in direct motion vector prediction mode. In direct motion vector prediction mode, the motion vector component differences are both set to zero; hence the motion vector components used for the inter prediction process are equal to the motion vector predictor components, MvL0[i][j][k] = MvpL0[i][j][k] and MvL1[i][j][k] = MvpL1[i][j][k]. Reference picture indices are also computed using the direct reference picture index prediction. When direct_spatial_mv_pred_flag is equal to 1, the spatial direct motion vector and reference picture index prediction mode specified in subclause 8.4.1.4.2.2 is used.

When direct_spatial_mv_pred_flag is equal to 0, the temporal direct motion vector and reference picture index prediction mode specified in subclause 8.4.1.4.2.3 is used, which utilizes the information stored in the co -located macroblock specified in subclause 8.4.1.4.2.1.

8.4.1.4.2.1 Derivation process for the co-located macroblock

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify ou tput of this process]

[Ed. Note: Limin's text needs harmonization: For a macroblock in B slice, the co -located macroblock is defined as the macroblock resided in the same geometric location of the first reference picture in list 1. A macroblock and its co -located shall have the same horizontal coordinates. The vertical coordinates may be different, depending upon whether macroblocks reside in frame or field picture. Let $y_{current}$ and $y_{co\text{-}located}$ be the vertical coordinates of a macroblock and its co-located macroblock. If the current macroblock and its co -located macroblock both reside in frame picture or field picture, $y_{co\text{-}located} = y_{current}$. If the current macroblock is in frame picture and its co -located macroblock in field picture, $y_{co\text{-}located} = y_{current} / 2$. If the current macroblock is in field picture and its col -located macrblock in frame picture, $y_{co\text{-}located} = 2 \times y_{current}$.] [Ed. Note: The harmonization of this section should be done in conjunction with more precise macroblock numbering in MB -AFF in section 8.3.7.]

[Ed. Note: Jill, Alexis, Limin ... please fill in] [Ed. Note: Jill's note – is this good enough?][Ed. Note: Alexis – It is probably necessary to make clarification here for cases 2 and 3 for interlace. Possibly these cases could also be moved here][Ed. Note: Jill – Also need to distinguish between frame pictures in MB -AFF mode and those not in MB -AFF mode, since macroblock numbering is different in those two cases. Can Limin trying writing these?]

8.4.1.4.2.2 Derivation process for spatial direct luma mo tion vector and reference picture index prediction mode

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

When direct_spatial_mv_pred_flag is equal to 1, spatial direct mode is used to derive the luma motion vectors and reference picture indices for skipped macroblocks and macroblocks with mb_type Direct_16x16 or macroblock partitions with sub_mb_type Direct_8x8.

The first step in the Spatial direct luma moti on vector and reference picture index prediction mode is the determination of the reference picture index for each list (list 0 and list 1), RefIdxL0 [i] and RefIdxL0[i]. The same process is used for macroblocks coded using the Direct_16x16 macroblock type, and sub-macroblocks coded using the Direct_8x8 sub - macroblock type.

The reference picture indices for the neighbouring blocks A, B, C, and D within the current slice for the current luma macroblock E, for both Direct_16x16 macroblock and Direct_8x8 sub-macroblock types, as described in subclause 8.4.1.1 and shown in Figure 8-7, shall be used to determine the reference picture index for each list. As in 8.4.1.1, the reference index of block D is o nly used if block C is not available in the current slice. The reference picture index for each list (list 0 and list 1) shall be the minimum reference picture index among the reference picture indices used from the same list for the prediction or bi -prediction of the neighbouring blocks. If no neighbouring blocks are present within the current slice that use prediction from the same list, either for prediction or bi -prediction, the reference index for that list shall be inferred as not existing.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

If the reference index exists for either list, decision of the associated motion vector values for that list for each 4x4 block of the current macroblock depends on the coded parameters of the co-located 4x4 blocks in the first picture in list 1, if both pictures have been coded using the same picture type (frame or field). If the current picture is coded in field mode and the list 1 picture in frame mode, only the same parity lines and their associated motion parameters are considered. Otherwise, if current picture is in frame mode and the first list 1 picture in field mode, then the corresponding frame mode picture containing the first list 1 picture is considered, if such exists. If the reference index of either list for the current block is zero, the associated list 1 picture is a short-term picture, and if all lines of the co-located 4x4 block exist and were predicted using list 0 prediction, reference picture index 0 and motion vector components in the range of -1 to 1 in sub-pixel units, inclusive, then the motion vector for this 4x4 block is set to (0, 0). Otherwise, the associated candidate motion vector for each list shall be obtained, using the 16x16 block motion vector prediction, as described in subclause 8.4.1.2.2 by using the reference picture index of each list.

If both candidate reference picture indices exist, then the block is predicted as a bi-prediction block using the corresponding reference picture index and associated motion vector for each list. Otherwise, if a reference picture index exists for only one of the two lists, the block shall be predicted by single-list prediction using the reference picture index and associated motion vector for the existing list. Finally, if neither reference picture index exists, bi-prediction shall be used with reference picture index zero and associated motion vector (0, 0) for both lists.

8.4.1.4.2.3 Derivation process for temporal direct luma motion vector and reference picture index prediction mode

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

When direct_spatial_mv_pred_flag is equal to 0, temporal direct mode is used to derive the luma motion vectors and reference picture indices for skipped macroblocks and macroblocks with mbtype Direct_16x16 or macroblock partitions with sub_mbtype Direct_8x8.

[Ed. Note: added the following paragraph and Figure from Lowell. Needs to be incorporated ...]

In the temporal direct mode, if the entire macroblock is skipped or coded using the Direct_16x16 macroblock type, then the same macroblock structure [Ed. Note: is "structure" defined? Should a different term be used?] as for the co-located macroblock defined in 8.4.1.4.2.1 is used. If direct_temporal_constrained_flag is equal to 0, then the same 8x8 sub-macroblock partitioning as in the co-located sub-macroblock is used, otherwise the 8x8 sub-macroblocks are constrained to not be subdivided. That is, if direct_temporal_constrained_flag is set to 0, then for each block of the current sub-macroblock, the list 0 and list 1 motion vectors are computed as scaled versions of the stored motion vector of the colocated block in the list 1 reference picture as described below. If direct_temporal_constrained_flag is set to 1, then for each 8x8 sub-macroblock of the current macroblock, only the motion vectors of the corner 4x4 block of the 8x8 sub-macroblock are stored.

[Ed. Note: The corner 4x4 block of an 8x8 submacroblock is defined to be the 4x4 block with the same relative decode number (see **Error! Reference source not found.**) as the 8x8 submacroblock in which it is contained (see Figure 8-10). For example, the lower-left 4x4 block within the lower-left 8x8 submacroblock is its corner 4x4 block. As a further example, the motion vectors of the lower-right 8x8 submacroblock are computed as scaled versions of the list 0 motion vectors of the lower-right 4x4 block that is contained within the co-located macroblock's lower-right 8x8 submacroblock.]

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

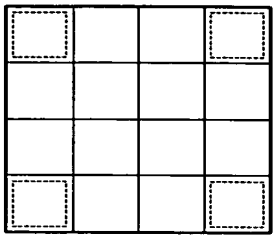


Figure 8-10 – Double-dashed border to indicate each 'corner' 4x4 block of each co-located 8x8 submacroblock, from which each temporal direct-mode 8x8 sub-macroblock derives its motion vectors if direct_temporal_constrained_flag is set to 1.

In temporal direct mode, the same frame or field decoding mode as for the co-located macroblock of the first picture (the picture at index 0) in list 1 is used. For each block of the current macroblock, the list 0 and list 1 motion vectors are computed as scaled versions of the stored motion vector of the co-located block in the list 1 reference picture as described below.

The list 0 reference picture for the direct mode current block is the same as the stored reference picture used for the co-located block. When temporal direct mode is used, the motion vector, frame or field decoding mode, and reference picture index must be stored for each sub macrobloc k. If a list 0 prediction is used, either alone or as part of bi-prediction, the list 0 motion vector and reference picture index are stored. If a list 0 prediction is not used, and a list 1 prediction is used, the list 1 motion vector and reference pict ure index are stored. If neither a list 0 or list 1 prediction is used, as is the case for an intra-coded block, a zero motion vector and ref_idx 0 of list 0 are stored. The list 0 and list 1 motion vectors for direct mode blocks are calculated differentl y depending on whether pic_struct and the reference indicate fields or frames. [Ed. Note: This was covered already in 8.4.1.]

If the co-located block is in a long-term reference picture or if the co-located block refers to a long-term picture, the list 0 and list 1 motion vectors for the direct mode current block are calculated by:

$$MV_0 = MV$$

$$MV_1 = 0$$

It is forbidden for direct mode to be used if the stored reference picture of the co-located block is not in the current picture's reference picture list.

[Ed. Note: Limin had provided some alternate text for the 3 cases that hasn't been integrated here.] [Ed. Note: I don't like using the word "forbidden". Should rephrase to something like "direct mode shall not be used...".]With possible adaptive switching of frame/field coding at the picture level and/or at the macroblock level, the current block or its co-located block can be coded in either frame structure or field structure. Hence, there are three cases that must be considered based on possible combinations of frame or field coding for a pair of a macroblocks in the current B slice and its co-located macroblock. Calculations of the two motion vectors in direct mode are slightly different for the three cases.

Case 1: The current macroblock and its co-located macroblock are either both in frame mode or both in field mode, as shown in Figure 8-11 and Figure 8-12. The list 0 reference for each block within the current macroblock is the same as the stored reference of the co-located block. Two motion vectors (MV_0, MV_1) are calculated by:

$$Z = (TD_B \times 256) / TD_D \qquad MV_0 = (Z \times MV + 128) >> 8$$

$$W = Z - 256 \qquad MV_1 = (W \times MV + 128) >> 8$$

where MV is the stored motion vector of the co-located block in list 1 reference picture. TD_B is the DiffPicOrderCount between the current picture and the list 0 reference picture, DiffPicOrderCount(ref_idx_l0, current_pic) and TD_D is the

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

DiffPicOrderCount between the list 1 reference picture and the list 0 reference picture (see **Error! Reference source not found.**), [Ed. Note: missing cross-reference]DiffPicOrderCount(ref_idx_l0, ref_idx_l1).

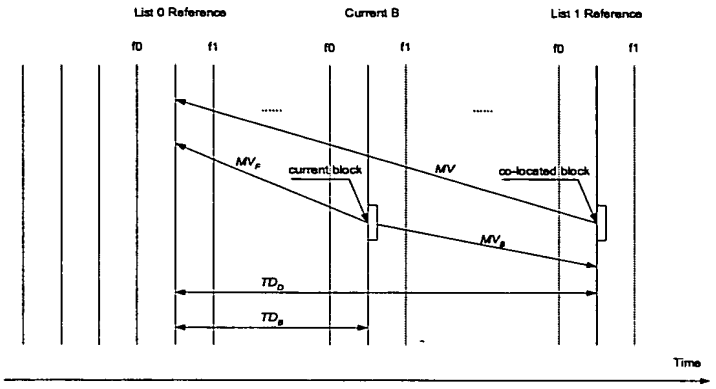


Figure 8-11 – Both the current block and its co-located block in the list 1 reference picture are in frame mode (f0 and f1 indicate the corresponding fields)

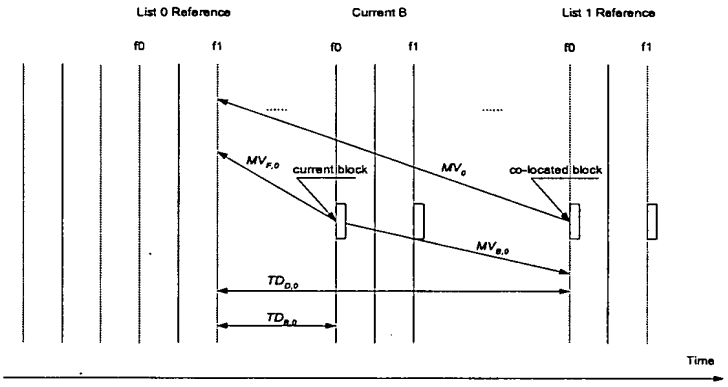


Figure 8-12 – Both the current macroblock and its co-located macroblock in the temporally subsequent picture are in field mode.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

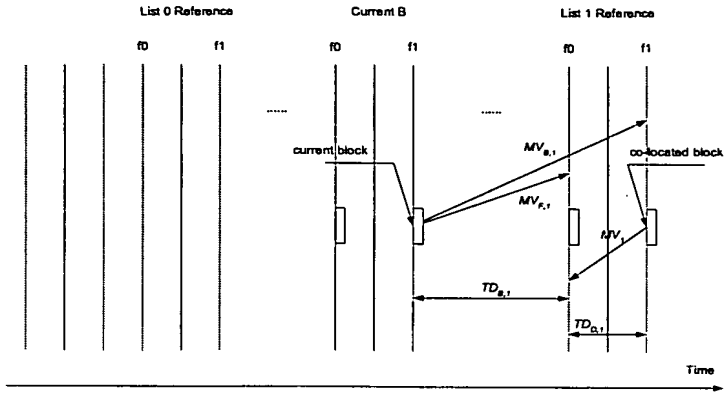


Figure 8-13 – The list 0 motion vector of the co-located block in field 1 of the list 1 reference frame may point to field 0 of the same frame.

Case 2: The current block is in field mode and its co-located block is in frame mode, as shown in Figure 8-14. Let $y_{current}$ and $y_{co\text{-}located}$ be the vertical indices of the blocks in the current macroblock within the field and in its co-located macroblock within the frame respectively. Then, $y_{co\text{-}located} = 2 \times y_{current}$. The blocks in the current macroblock and its co-located macroblock have the same horizontal indices. If the reference pictures are in their default order, and have not been reordered, the list 0 reference field is the same-parity field of the list 0 frame, and the list 1 reference field will be the same-parity field of the list 1 reference frame as shown in Figure 8-15.

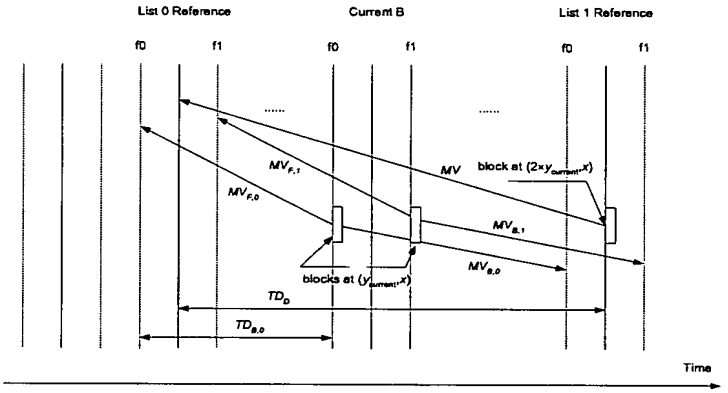


Figure 8-14 – The current macroblock is in field mode and its co-located macroblock in the list 1 reference picture is in frame mode

The motion vectors of a direct mode block are calculated from the list 0 motion vector of the co-located block in the list1 reference frame as follows:

$Z = (TD_B \times 256) / TD_D$ $\quad MV_0 = (Z \times MV + 128) >> 8$

$W = Z - 256$ $\quad MV_1 = (W \times MV + 128) >> 8$

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

where MV is the list 0 motion vector of the co-located block in the list 1 reference picture with the vertical component shifted right by 1, because of the difference in vertical distance between fields and frames. TD_B is the DiffPicOrderCount between the current field and the list 0 reference frame, DiffPicOrderCount(ref_idx_l0, current_field) and TD_D is the DiffPicOrderCount between the list 1 reference frame and the list 0 reference field (see **Error! Reference source not found.**), [Ed. Note: missing cross-reference]DiffPicOrderCount(ref_idx_l0, ref_idx_l1).

Case 3: The current macroblock is in frame mode while its co-located macroblock in the list 1 reference picture is in field mode, as shown in Figure 8-15. Let $y_{current}$ and $y_{co\text{-}located}$ be the vertical indices of the blocks in the current macroblock within the frame and in its co-located macroblock within the field respectively. Then, $y_{co\text{-}located} = y_{current} / 2$. The blocks in the current macroblock and in its co-located macroblock have the same horizontal indices. The two fields of the co-located block in the list 1 reference may be coded in different modes. The default reference picture ordering is shown in Figure 8-15 [Ed. Note: I'm not too confident about this last sentence.].

The motion vectors of the direct mode block are calculated as follows:

$$Z = TD_B \times 256 / TD_D \qquad MV_0 = (Z \times MV + 128) >> 8$$

$$W = Z - 256 \qquad MV_1 = -(W \times MV + 128) >> 8$$

where MV is the list 0 motion vector of the co-located block in the list 1 reference picture with its vertical component shifted left by 1, because of the difference in vertical distance between fields and frames

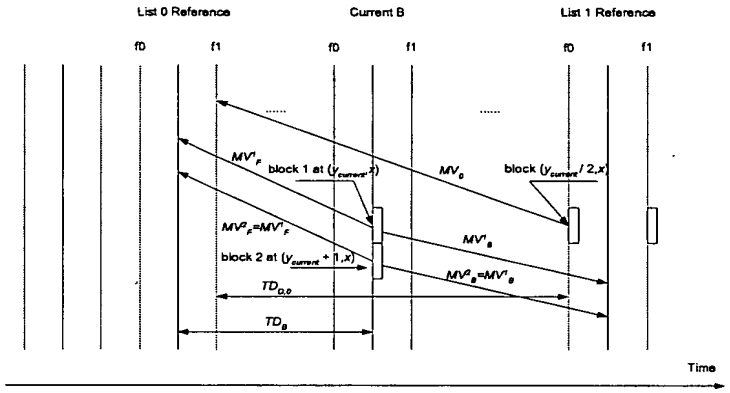


Figure 8-15 – The current macroblock is in frame mode while its co-located macroblock in the list 1 reference picture is in field mode.

8.4.1.5 Calculation of the resulting luma motion vector

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

[Ed. Note: This whole section's concepts are already addressed in other sections 8.4.1 and 8.3.7, and sbouldn't really be in both places. Which place should they be?]

For skipped macroblocks and for macroblocks coded in Direct_16x16 macroblock type or for sub-macroblocks coded in Direct_8x8 sub-macroblock type, the differential motion vector components are set to 0, so the motion vector used for inter prediction is equal to the motion vector predictor, MvL0[i][j][k] = MvpL0[i][j][k] and MvL1[i][j][k] = MvpL1[i][j][k]. [Ed. Note: This concept is already addressed in 8.4.1, and shouldn't really be in both places. Which place should it be?]

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

When decoding P and SP slices, the resulting motion vector components are obtained by adding the prediction of the motion vector components to the differential motion vector components.

MvL0[i][j][k] = MvpL0[i][j][k] + mvd_l0[i][j][k]

MvL1[i][j][k] = MvpL1[i][j][k] + mvd_l1[i][j][k]

[Ed. Note: This concept is already addressed in 8.4.1, a nd shouldn't really be in both places. Which place should it be?]

When decoding B slices, the motion vector predictor is formed using direct mode, using spatial direct mode if direct_spatial_mv_pred_flag equals 0 and temporal direct mode is direct_spatial _mv_pred_flag equals 1. No differential motion vector is decoded, so MvL0[i][j][k] = MvpL0[i][j][k] and MvL1[i][j][k] = MvpL1[i][j][k].

[Ed. Note: I think this whole paragraph was covered in 8.3.7]

8.4.1.6 [Ed, Note: This whole paragraph is covered in 8.4.1]Derivation process for chroma motion vectors

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

Chroma motion vectors are derived from the luma motion vectors. Since the accuracy of luma motion vectors is 1/4 pel and chroma has half resolution compared to luma, the accuracy of chroma motion vectors is 1/8 pel. Due to the lower resolution of the chroma array relative to the luma array, a chroma motion vector applies to 1/4 as many samples as the luma motion vector.

NOTE - For example if the luma vector applies to 8x16 luma samples, the corresponding chroma vector applies to 4x8 chroma samples and if the luma vector applies to 4x4 luma samples, the corresponding chroma vector applies to 2x2 chroma samples.

In frame coding mode, the horizontal and vertical components of the chroma motion vector are obtained by dividing the corresponding components of luma motion vector by 2.

In field decoding mode, the horizontal component of the chroma motion vector is obtained by dividing the horizontal component of the corresponding luma motion vector by 2. The vertical component of the chroma motion vector is dependent on the parities of the current fie ld and the reference field where the corresponding luma motion vector is defined. The vertical component of chroma motion vector (MVCy) is calculated from the vertical component of the luma motion vector (MVy) according to the Table 8-8. Here, the units of MVy and MVCy represent full-sample vertical displacement units in luma and chroma respectively.

Table 8-8 – Calculation of vertical component of chroma vector in field coding mode

Parity conditions		Vertical component of chroma motion vector
Reference field	Current field	
Top field	Bottom field	$MVCy = \frac{MVy}{2} + \frac{1}{4}$
Bottom field	Top field	$MVCy = \frac{MVy}{2} - \frac{1}{4}$
Otherwise		$MVCy = \frac{MVy}{2}$

MVy: Vertical component of luma motion vector

MVCy: Vertical component of chroma motion vector

NOTE – Chroma motion vectors are exactly parallel to the corresponding luma motion vectors in both frame coding mode and field coding mode. In field coding mode, the locations of chroma samples in relative to luma samples in a top field is different from those in bottom field (see subclause 6.2 and Figure 6-2). From this reason, MVCy is shifted up/down by 1/4 pel from the MVy / 2 when vectors are defined between fields with different parity.

8.4.2 Decoding process for inter prediction samples

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Given the motion vectors, the prediction of the samples for the macroblock or its partitions is calculated. If weighted prediction is used, a weighting factor and/or offset are applied to the sample prediction.

If the mbtype of the macroblock or macroblock partition is Pred_L0 or Pred_L1, or for skippe d macroblocks in P or SP slices, a single predictor is formed. If the mbtype of the macroblock or macroblock partition is BiPred or Direct, or for skipped macroblocks in B slices, two predictors are formed, which are combined into a single predictor.

To obtain the prediction of the macroblock or partition sample values, the following steps are conducted

- reference picture selection
- interpolation process for fractional sample motion vectors to form a sample predictor
- possible weighting or combining of samp le predictors to form the final sample predictor

8.4.2.1 Reference picture selection process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

When mb_adaptive_fram e_field_flag is equal to 0, the reference picture selection is described in subclauses ??? [Ed. Note: ??]. If field_pic_flag indicates a field picture, only the reference field indicated by the ref_idx_l0 or ref_idx_l1 is used in the inter prediction. The motion vectors to be used are described in subclauses 7.3.5.1-7.3.5.2, 7.4.5, 7.4.6.1??? [Ed. Note: ??] and **Error! Reference source not found.** [Ed. Note: ??].

If the current macroblock pair is in frame decoding mode and one or more neighbouring blocks is in field mode, the reference frame index used in MV prediction from the field coded neighbours is obtained by dividing the reference field list index by 2 and truncation of any fractional result toward zero to obtain the effective frame index for prediction. If the macroblock pair is in frame mode, the reference field number for any frame coded neighbour is obtained by multiplying the reference frame index by 2. [Ed.Note: Some wording problems here] [Ed.Note: Limin: This paragraph is not correct, and does not belong here. For ref picture number selection for PMV, refer top 8.4.1.2.1.]

When a macroblock pair is in field mode, each field macroblock ma y refer to any (top or bottom) field in the decoded picture buffer.

[Ed.Note: Needs much more details]

8.4.2.2 Fractional sample interpolation process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed. Note: specify output of this process]

Quarter-sample interpolation with a 6-tap filter is applied to the luma samples in the block as described in subclauses 8.4.2.2.1. The interpolation process for fractional chroma samples in both cases is described in subclause 8.4.2.2.2.

If a sample referred to in the interpolation process (necessarily integer accuracy) is outside of the reference picture it shall be replaced by the nearest sample belonging to the picture (an edge or corner sample). Motion vectors are allowed to point to samples outside the reference picture. If a sample outside the reference picture is referred to in the prediction process, the nearest sample belonging to the picture (an edge or c orner sample) shall be used.

8.4.2.2.1 Luma sample interpolation process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

In Figure 8-16, the positions labelled with upper-case letters within shaded blocks represent reference picture samples at integer sample positions, and the positions labelled with lower -case letters within un-shaded blocks represent reference picture samples at fractional sample positions.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

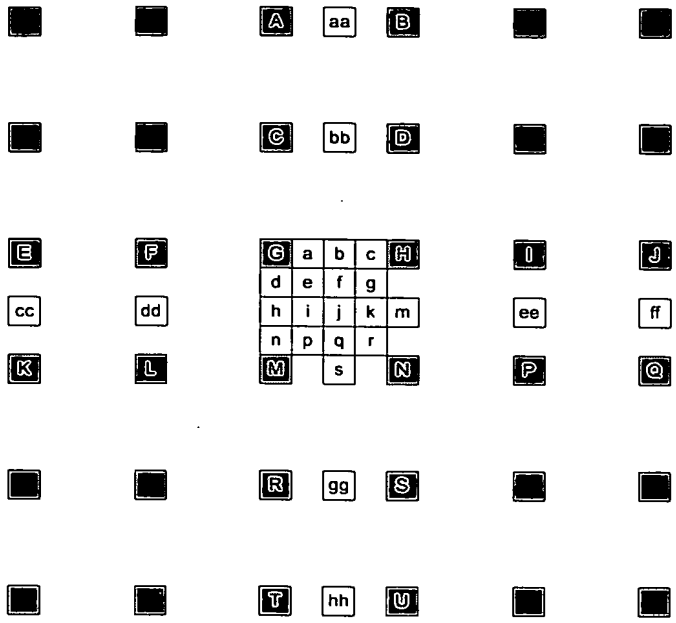


Figure 8-16 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation.

The luma prediction values at half sample positions shall be obtained by applying a 6 -tap filter with tap values (1, -5, 20, 20, -5, 1). The luma prediction values at quarter sample positions shall be obtained by averaging samples at integer and half sample positions. The process for each fractional position is described below.

- The samples at half sample positions labelled 'b' shall be obtained by first calculating intermediate values denoted as '*b*' by applying the 6-tap filter to the nearest integer position samples in the horizontal direction. The samples at half sample positions labelled 'h' shall be obtained by first calculating intermediate values denoted as '*h*' by applying the 6-tap filter to the nearest integer position samples in the vertical direction:

  b=(E-5F+20G+20H-5I+J),

  h=(A-5C+20G+20M-5R+T).

  The final prediction values shall be calculated using:

  b=Clip1((b+16)>>5),

  h= Clip1((h+16)>>5).

- The samples at half sample position labelled as 'j' shall be obtained by first calcul ating intermediate value denoted as '*j*' by applying the 6-tap filter to the intermediate values of the closest half sample positions in either the horizontal or vertical direction because these yield an equivalent result.

  j=*cc*-5*dd*+20h+20m-5*ee*+*ff*, or

  j=*aa*-5*bb*+20b+20s-5*gg*+*hh*,

  where intermediate values denoted as '*aa*', '*bb*', '*gg*', '*s*' and '*hh*' shall be obtained by applying the 6-tap filter horizontally in an equivalent manner to '*b*' and intermediate values denoted as '*cc*', '*dd*', '*ee*', '*m*' and '*ff*' shall be obtained by applying the 6-tap filter vertically in an equivalent manner to '*h*'. The final prediction value shall be calculated using: j=Clip1((j+512)>>10).

- The samples at quarter sample positions labelled as 'a', 'c', 'd', 'n', 'f', 'i', 'k' and 'q ' shall be obtained by averaging with truncation the two nearest samples at integer and half sample positions using:

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

a=(G+b)>>1, c=(H+b)>>1, d=(G+h)>>1, n=(M+h)>>1, f=(b+j)>>1, i=(h+j)>>1, k=(j+m)>>1 and q=(j+s)>>1.

- The samples at quarter sample positions labelled as 'e', 'g', 'p', and 'r' shall be obtained by averaging with truncation the two nearest samples at half sample positions in the diagonal direction using e=(b+h)>>1, g=(b+m)>>1, p=(h+s)>>1, and r=(m+s)>>1.

8.4.2.2.2 Chroma sample interpolation process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

Inter chroma prediction values at fractional sample positions shall be obtained using:

$$v = ((8 - d_x) \cdot (8 - d_y) \cdot A + d_x \cdot (8 - d_y) \cdot B + (8 - d_x) \cdot d_y \cdot C + d_x \cdot d_y \cdot D + 32) >> 6 \qquad (8\text{-}32)$$

where A, B, C and D are the integer position reference picture samples surrounding the fractional sample location; d_x and d_y are the fractional parts of the sample position in units of one eighth samp le. The relationships between the variables in Equation 8-32 and reference picture positions are illustrated in Figure 8-17.

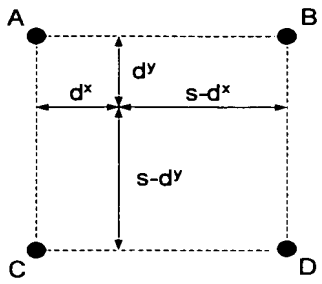


Figure 8-17 – Fractional sample position dependent variables in chroma interpolation and surrounding integer position samples A, B, C, and D. [Ed. Note: Edit to replace s with 8 and d^x and d^y with d_x and d_y]

8.4.2.3 Weighted sample prediction process

Input to this process is/are ??? [Ed.Note: specify inp ut to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

For P and SP slices, if weighted_pred_flag is equal to 0 the default weighted sample prediction process as described in subclause 8.4.2.3.1 is used for macroblocks or partitions coded using Pred_L0 and Pred_L1 prediction.

For P and SP slices, if weighted_pred_flag is equal to 1, explicit weighted prediction is applied to P and SP slices as described in subclause 8.4.2.3.2.

For B slices, if weighted_bipred_explicit flag is equal to 0 and weighted_bipred_implicit_flag is equal to 0, the default weighted sample prediction process as described in subclause 8.4.2.3.1 is used.

For B slices, if weighted_bipred_explicit flag is equal to 1, weighted sample prediction is used to form the final sample predictor, as described in subclause 8.4.2.3.2.

For B slices, if weighted_bipred_implicit_flag is equal to 1, the de fault weighted sample prediction process as described in subclause 8.4.2.3.1 is used for macroblocks or partitions coded using Pred_L0 and Pred_L1 prediction, and weighted sample prediction as described in subclause 8.4.2.3.2 is used for skipped macroblocks or macroblock or paritions coded using Bipred or Direct prediction.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

8.4.2.3.1 Default weighted sample prediction process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

For macroblocks or partitions coded using Pred_L0 and Pred_L1 prediction, the default weighting applied to all interpolated luma and chroma samples is equal to 1.

For macroblocks or partitions coded using bi-prediction, including skipped macroblocks and direct mode macroblocks or partitions, if weighted_bipred_explicit_flag is equal to zero and weighted_bipred_implicit_flag is equal to zero, the two predictions are combined into a single predict or using the following formula.

$$P = (P_0 + P_1 + 1) >> 1$$

where

P_0 = list 0 reference prediction block [Ed. Note: Not a defined term. What does it mean?]

P_1 = list 1 reference prediction block

If either weighted_bipred_explicit_flag or weighted_bipred_imp licit_flag is equal to 1, a weighted prediction is formed as described in subclause 8.4.2.3.2.

8.4.2.3.2 Weighted sample prediction process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

In P slices when weighted_pred_flag is equal to 1, or in B slices, when weighted_bipred_explicit_flag is equal to one , weighted prediction is applied for all non-intra macroblocks and partitions, includin g those coded using Pred_L0, lPred_L1, Bipred, skipped and Direct prediction. When weighted_bipred_implicit_flag is equal to one, weighted prediction is applied to all bi-prediction macroblocks and partitions, including skipped macroblocks and those coded using Bipred, or Direct prediction.

For reference list 0 prediction, the luma prediction is generated as

$$P = \mathrm{Clip1}(((P_0 \cdot W_0 + 2^{LWD-1}) >> LWD) + O_0) \quad (8\text{-}33)$$

For reference list 1 prediction, the luma prediction block is generated as:

$$P = \mathrm{Clip1}(((P_1 \cdot W_1 + 2^{LWD-1}) >> LWD) + O_1) \quad (8\text{-}34)$$

where

P_0 = reference prediction block from list 0

P_1 = reference prediction block from list 1

LWD = luma_log_weight_denom

W_0 = luma_weight_l0[ref_idx_l0]

W_1 = luma_weight_l1[ref_idx_l1]

O_0 = luma_offset_l0[ref_idx _l0]

O_1 = luma_offset_l1[ref_idx_l1]

When bi-prediction is used, the luma prediction block is generated as:

$$P = \mathrm{Clip1}\ (((P_0\, W_0 + 2^{LWD-1}) >> LWD)\ + O_0 + ((P_1\, W_1 + 2^{LWD-1}) >> \mathrm{LWD}) + O_1 + 1) >> 1) \quad (8\text{-}35)$$

If weighted_bipred_implicit_flag is equal to one,

$LWD = 7$

$O0 = 0$

$O1 = 0$

$W0$ = *(128* DiffPicOrderCount(ref_idx_l1, current_pic) / DiffPicOrderCount (ref_idx_l1, ref_idx_l0) = W>>1 of subclause 8.4.1.2.2*

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

$W1$ = (128* DiffPicOrderCount(current_pic,ref_idx_l0) / DiffPicOrderCount(ref_idx_l1 ,ref_idx_l0) = Z>>1 of subclause 8.4.1.2.2

except if DiffPicOrderCount(ref_idx_l1,ref_idx_l0) is equal to zero or if one or both reference pictures is a long term reference picture, in which case, $W_0 = W_1 = 128$.

Otherwise if weighted_bipred_explicit_flag is equal to one and use_implicit_weights_for_bipred_flag is equal to zero

LWD = luma_log_weight_denom

W_0 = luma_weight_l0[ref_idx_l0]

W_1 = luma_weight_l1 [ref_idx_l1]

O_0 = luma_offset_l0[ref_idx_l0]

O_1 = luma_offset_l1[ref_idx_l1]

For reference list 0 prediction, the chroma prediction block is generated as

$$P = \text{Clip1}(128 + (((CP_0 - 128) \cdot CW_0 + 2^{CWD-1}) >> CWD) + CO_0) \quad (8\text{-}36)$$

For reference list 1 prediction, the chroma prediction block is generated as:

$$P = \text{Clip1}(128 + (((CP_1 - 128) \cdot CW_1 + 2^{CWD-1}) >> CWD) + CO_1) \quad (8\text{-}37)$$

When bi-prediction is used, the chroma prediction block is generated as

$$P = \text{Clip1}(((((CP_0 - 128)\ CW_0 + 2^{CWD-1}) >> CWD) + CO_0 + (((CP_1 - 128)\ CW_1 + 2^{CWD-1}) >> CWD) + CO_1 + 1) >> 1) \quad (8\text{-}38)$$

where

j = 0 for Cb and 1 for Cr

CP_0 = chroma reference prediction block from list 0

CP_1 = chroma reference prediction block from list 1

If weighted_bipred_implicit_flag is equal to one,

$CWD = 7$

$CO0 = 0$

$CO1 = 0$

CW_0 = (128* DiffPicOrderCount(ref_idx_l1, current_pic) / DiffPicOrderCount (ref_idx_l1, ref_idx_l0) = W>>1 of subclause 8.4.1.2.2

CW_1 = (128* DiffPicOrderCount(current_pic,ref_idx_l0) / DiffPicOrderCount(ref_idx_ l1 ,ref_idx_l0) = Z>>1 of subclause 8.4.1.2.2

except if DiffPicOrderCount (ref_idx_l1, ref_idx_l0) is equal to zero or if one or both reference pictures is a long term reference, in which case, $CW_0 = CW_1 = 128$.

Otherwise if weighted_bipred_explicit_fla g is equal to one and use_implicit_weights_for_bipred_flag is equal to zero

CWD = chroma_log_weight_denom

CW_0 = chroma_weight_l0[ref_idx_l0][j]

CW_1 = chroma_weight_l1 [ref_idx_l1][j]

CO_0 = chroma_offset_l0[ref_idx _l0][j]

CO_1 = chroma_offset_l1[ref_idx_l1][j]

To limit the calculation to 16-bit precision, the following conditions shall be met:

$-128 \le W_0 \le 128$

$-128 \le W1 \le 128$

$-128 \le CW0 \le 128$

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

$-128 \leq CW1 \leq 128$

8.5 Transform coefficient decoding process and picture construction process prior to deblocking filter process

Input to this process are mb_type, Intra16x16DCLevel (if available), Intra16x16ACLevel (if available), LumaLevel (if available), ChromaDCLevel, and ChromaACLevel. [Ed.Note: further specify the use of these variables.]

Output of this process are the residual samples of the macroblock.

This subclause specifies transform coefficient decoding and picture construction prior to the deblocking filter process.

8.5.1 Zig-zag scanning process and field scanning process

Input to this process is an 1-D array of levels.

Output of this process is a 2-D array of levels assigned to locations in the transform block.

[Ed.Note: we need to be more specific here. This is not easy ... esp. concerning Intra_16x16 ...]

The decoder maps the sequence of transform coefficient levels to the transform coefficient level positions. For this mapping, the two scanning pattern shown in Figure 8-18 are used. The zig-zag scan shall be used for blocks decoded in frame mode, and the field scan shall be used for blocks decoded in field mode.

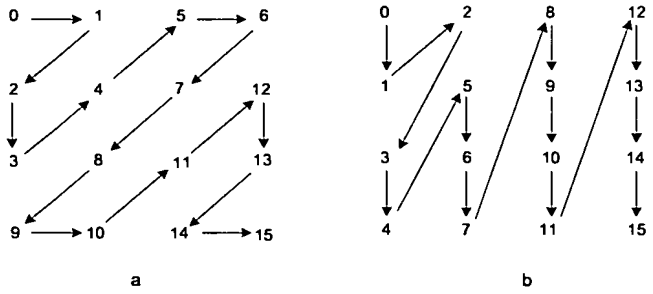


Figure 8-18 – a) Zig-zag scan. b) Field scan

When decoding a macroblock coded using Intra_16x16 macroblock prediction mode, the coefficients of the 4x4 luma DC transform are scanned in the same scan order as luma 4x4 coefficient blocks. Then for each 4x4 block of luma coefficients with AC coefficients to scan, the 15 remaining coefficients are scanned by starting the zig-zag scan in frame mode or field scan in field mode at its second position.

The coefficients of the 2x2 chroma DC transform are scanned in raster order. Then for each 4x4 block of chroma coefficients with AC coefficients to scan, the 15 remaining coefficients are scanned by starting the zig-zag scan in frame mode or the field scan in field mode at its second position.

8.5.2 Scaling and transformation process

Inputs to this process are

- QP_Y;
- QS;
- parsed transform coefficient level values;
- chroma_qp_index_offset.

Outputs of this process are:

- QP_C;
- QS_C.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

- sample residual blocks X'';

QP scaling parameter values QP_Y, QP_C, QP_I, QS, and QS_C have a range from 0 to 51, inclusive. The value of QP_C for chroma is determined from the current value of QP_Y and the value of chroma_qp_index_offset. The sc aling equations are specified such that the equivalent scaling parameter doubles for every increment of 6 in QP. Thus, there is an increase in scaling magnitude of approximately 12% for each increase of 1 in the value of QP.

The value of QP_C shall be determined as specifed in Table 8-9 based on the indexing QP, denoted QP_I. The value of QP_I shall be calculated as specified by

$$QP_I = \text{Clip3}(0, 51, QP_Y + \text{chroma\_qp\_index\_offset}) \tag{8-39}$$

Table 8-9 – Specification of QP_C as a function of QP_I

QP_I	<30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
QP_C	$=QP_I$	29	30	31	32	32	33	34	34	35	35	36	36	37	37	37	38	38	38	39	39	39	39

In the case of SP and SI slices, QS_C is calculated using the above process, substituting all QP values with equivalent QS values.

QP_Y shall be used as the QP to be applied for luma scaling and QP_C shall be used for chroma scaling.

The coefficients $R_{ij}^{(m)}$ specified as follows are used in Equations 8-43, 8-44, 8-46, 8-47, 8-48, 8-62, and 8-67:

$$R_{ij}^{(m)} = \begin{cases} V_{m0} & \text{for } (i,j) \in \{(0,0),(0,2),(2,0),(2,2)\}, \\ V_{m1} & \text{for } (i,j) \in \{(1,1),(1,3),(3,1),(3,3)\}, \\ V_{m2} & \text{otherwise;} \end{cases} \tag{8-40}$$

where the first and second subscripts of V are row and column indices, respectively, of the matrix specified as:

$$V = \begin{bmatrix} 10 & 16 & 13 \\ 11 & 18 & 14 \\ 13 & 20 & 16 \\ 14 & 23 & 18 \\ 16 & 25 & 20 \\ 18 & 29 & 23 \end{bmatrix}. \tag{8-41}$$

8.5.2.1 Scaling and transformation process for luma DC coefficients for Intra_16x16 macroblock type

Inputs to this process are

- QP_Y;
- parsed transform coefficient level values for luma DC coefficients of Intra_16x16 macroblocks;

Outputs of this process are:

- 16 scaled DC values for luma 4x4 blocks of Intra_16x16 macroblocks;

After decoding the coefficient levels for a 4x4 block of luma DC coefficients coded in 16x16 intra mode and assembling these into a 4x4 matrix C of elements c_{ij}, a transform process shall be applied in a manner mathematically equivalent to the following process. The process uses application of a transform before the scaling process.

The transform for the 4x4 luma DC coefficients in 16x16 intra macroblocks is specified by:

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}. \quad (8\text{-}42)$$

A bitstream conforming to this Recommendation | International Standard shall not contain data that results in any element of F that exceeds the range of integer values from -2^{15} to $2^{15}-1$, inclusive.

After the transform, scaling is performed according to the following:

a) If QP is greater than or equal to 12, then the scaled result shall be calculated as

$$DC_{ij} = [F_{ij} \cdot R_{00}^{(QP\%6)}] << (QP/6 - 2), \quad i,j = 0,\ldots,3. \quad (8\text{-}43)$$

b) If QP is less than 12, then the scaled results shall be ca lculated as

$$DC_{ij} = [F_{ij} \cdot R_{00}^{(QP\%6)} + 2^{1-QP/6}] >> (2 - QP/6), \quad i,j = 0,\ldots,3. \quad (8\text{-}44)$$

A bitstream conforming to this Recommendation | International Standard shall not contain data that results in element of DC_{ij} that exceeds the range of integer values from -2^{15} to $2^{15}-1$, inclusive.

8.5.2.2 Scaling and transformation process for chroma DC coefficients

Inputs to this process are

- QP_C;
- parsed transform coefficient level values for chroma DC coefficients of a macroblock;

Outputs of this process are:

- 4 scaled DC values for chroma 4 x4 blocks of a macroblock;

After decoding the coefficient levels for a 2x2 block of chroma DC coefficients and assembling these into a 2x2 matrix C of elements c_{ij}, the transform process is applied before the scaling process.

Definition of transform:

$$F = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (8\text{-}45)$$

A bitstream conforming to this Recommendation | International Standard shall not contain data that results in any element of F that exceeds the range of integer values from -2^{15} to $2^{15}-1$, inclusive.

After the transform, scaling is performed according to the following.

a) If QP is greater than or equal to 6, then the scaling result shall be calculated as

$$DC_{ij} = [F_{ij} \cdot R_{00}^{(QP\%6)}] << (QP/6 - 1), \quad i,j = 0,1. \quad (8\text{-}46)$$

b) If QP is less than 6, then the scaling results shall be calculated by

$$DC_{ij} = [F_{ij} \cdot R_{00}^{(QP\%6)}] >> 1, \quad i,j = 0,1. \quad (8\text{-}47)$$

A bitstream conforming to this Recommendation | International Standard shall not contain data that results in any element of DC_{ij} that exceeds the range of integer values from -2^{15} to $2^{15}-1$, inclusive.

8.5.2.3 Scaling and transformation process for residual 4x4 blocks

Inputs to this process are

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

- QP_Y if a luma 4x4 block;
- QP_C if a chroma 4x4 block;
- parsed transform coefficient level values for a 4x4 residual block;
- DC value if a luma 4x4 block of an Intra_16x16 macroblock;
- DC value if a chroma 4x4 block;

Outputs of this process are:

- reconstructed sample residual 4x4 blocks X'';

Scaling of 4x4 block coefficient levels c_{ij} other than those as specified in subclauses 8.5.2.1 and 8.5.2.2 shall be performed according to

$$w_{ij} = [c_{ij} \cdot R_{ij}^{(QP\%6)}] << (QP/6), \quad i,j = 0,...,3 . \tag{8-48}$$

A bitstream conforming to this Recommendation | International Standard shall not contain data that results in a value of w_{ij} that exceeds the range of integer values from -2^{15} to $2^{15}-1$, inclusive.

After constructing an entire 4x4 block of scaled transform coefficients and assembling these into a 4x4 matrix W of elements w_{ij} illustrated as

$$W = \begin{bmatrix} w_{00} & w_{01} & w_{02} & w_{03} \\ w_{10} & w_{11} & w_{12} & w_{13} \\ w_{20} & w_{21} & w_{22} & w_{23} \\ w_{30} & w_{31} & w_{32} & w_{33} \end{bmatrix} \tag{8-49}$$

in which the w_{00} element may be a result DC_{ij} from Equation 8-43, 8-44, 8-46, or 8-47; or may be from Equation 8-48, as appropriate, the transform process shall convert the block of reconstructed transform coefficients to a block of output samples in a manner mathematically equivalent to the following process:

a) First, each row of reconstructed transform coefficients is transformed using a one -dimensional transform, and

b) Second, each column of the resulting matrix is transformed using the same one -dimensional transform.

The one-dimensional transform is specified as follows for four input samples w_0, w_1, w_2, w_3.

a) First, a set of intermediate values is computed:

$$z_0 = w_0 + w_2 \tag{8-50}$$

$$z_1 = w_0 - w_2 \tag{8-51}$$

$$z_2 = (w_1 >> 1) - w_3 \tag{8-52}$$

$$z_3 = w_1 + (w_3 >> 1) \tag{8-53}$$

b) Then the transformed result is computed from these intermediate values

$$x_0 = z_0 + z_3 \tag{8-54}$$

$$x_1 = z_1 + z_2 \tag{8-55}$$

$$x_2 = z_1 - z_2 \tag{8-56}$$

$$x_3 = z_0 - z_3 \tag{8-57}$$

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

A bitstream conforming to this Recommendation | International Standard shall not contain data that results in a value of z_0, z_1, z_2, z_3, x_0, x_1, x_2, or x_3 that exceeds the range of integer values from -2^{15} to $2^{15}-1$, inclusive, in either the first (horizontal) or second (vertical) stage of application of this transformation process. A bitstream conforming to this Recommendation | International Standard shall not contain data that results in a value of x_0, x_1, x_2, or x_3 that exceeds the range of integer values from -2^{15} to $2^{15}-33$, inclusive, in the second (vertical) stage of application of this transformation process.

After performing the transform in both the horizontal and vertical directions to produce a block of transformed samples,

$$X' = \begin{bmatrix} x'_{00} & x'_{01} & x'_{02} & x'_{03} \\ x'_{10} & x'_{11} & x'_{12} & x'_{13} \\ x'_{20} & x'_{21} & x'_{22} & x'_{23} \\ x'_{30} & x'_{31} & x'_{32} & x'_{33} \end{bmatrix}, \tag{8-58}$$

the final reconstructed sample residual values shall be obtained as

$$X''_{ij} = [x'_{ij} + 2^5] >> 6 \tag{8-59}$$

8.5.3 Picture construction process prior to deblocking filter process

Inputs to this process are

- constructed sample residual 4x4 blocks X'';
- prediction 4x4 blocks P;

Outputs of this process are:

- constructed sample residual blocks S' prior to deblocking filter operation;

Finally, the reconstructed sample residual values X'' from Equation 8-59 are added to the prediction values P_{ij} from inter prediction or spatial prediction and clipped to the range of 0 to 255 to form the final decoded sample result prior to application of the deblocking filter:

$$S'_{ij} = \text{Clip1}(P_{ij} + X''_{ij}) \tag{8-60}$$

[Ed.Note: add MB-AFF picture construction process here. Not sure here is exactly where it goes]

8.6 Decoding process for SP and SI slices

This process is invoked when decoding P macroblock types in an SP slice type or an SI macroblock type in SI slices.

Input to this process are the prediction residual transform levels and the predicted luma samples for the current macroblock.

Output of this process are the decoded samples of the current mac roblock prior to deblocking.

This subclause specifies the transform coefficient decoding process and picture construction process for non -intra macroblock types in SP slices and SI macroblock type in SI slices.

SP slices make use of inter predictive coding to exploit temporal redundancy in the sequence, in a similar manner to P slices. Unlike P slices, however, SP slice coding allows identical reconstruction of a slice even when different reference pictures are being used.

SI slices make use of spatial prediction, in a similar manner to I slices. SI slice coding allows identical reconstruction to a corresponding SP slice.

NOTE - Above mentioned properties of SP and SI slices provide functionalities for bitstream switching, splicing, random access, VCR functionalities such as fast-forward, and error resilience/recovery.

An SP slice consists of macroblocks coded either as I macroblock types or P macroblock types. An SI slice consists of macroblocks coded either as I macroblock types or SI macroblock type.

The transform coefficient decoding process and picture construction process prior to deblocking filter process for I macroblock types in SI slices shall be invoked specified in subclause 8.5. SI macroblock type shall be decoded as described below.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

8.6.1 SP decoding process for non-switching pictures

This process is invoked, when decoding P macroblock types in SP slices in which sp_for_switch_flag is equal to 0.

Input to this process are inter predicted samples for the current macroblock f rom subclause 8.4 and the prediction residual coefficient levels.

Output of this process are the decoded samples of the current macroblock prior to the deblocking filter process.

This subclause applies to all macroblocks in SP s lices in which sp_for_switch_flag is equal to 0, except those with macroblock prediction mode equal to Intra_4x4 or Intra_16x16. It does not apply to SI slices.

Figure 8-19 depicts generic decoding for non-intra coded macroblocks in SP slices. The prediction P_{ij} for the current macroblock of the slice being decoded is formed by the inter prediction block as specified in subclause 8.4. After forming the predicted block P_{ij}, decoding is performed as follows.

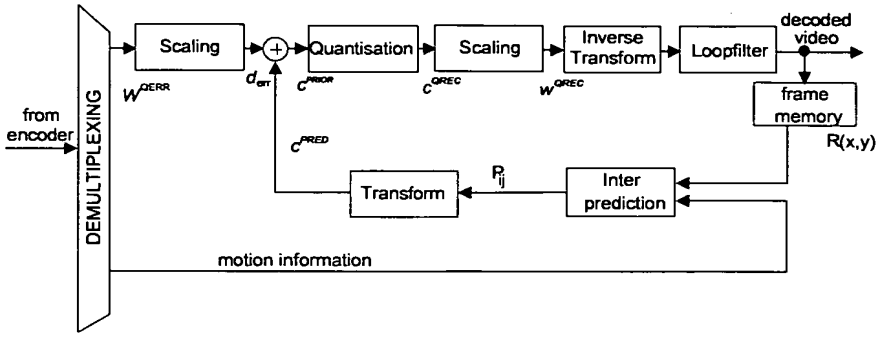


Figure 8-19 – A block diagram of a conceptual decoder for non-intra coded macroblocks in SP slices in which sp_for_switch_flag is equal to 0.

8.6.1.1 Luma transform coefficient decoding process

Input to this process are inter predicted luma samples for the current macroblock from subclause 8.4 and the prediction residual coefficient levels, LumaLevel.

Output of this process are the decoded luma samples of the c urrent macroblock prior to the deblocking filter process.

The predicted luma block, P_{ij}, where $P_{ij} = \{p_{00} \ldots p_{33}\}$, is transformed to produce transform coefficients c^{PRED} according to:

$$c^{PRED} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \quad (8\text{-}61)$$

The received prediction residual coefficients, w^{QERR}, are scaled using quantisation parameter QP, and added to the transform coefficients of the prediction block, as given by:

$$c_{ij}^{PRIOR} = c_{ij}^{PRED} + (((w_{ij}^{QERR} \; R_{ij}^{(QP\%6)} \; A_{ij}) << (QP/6)) >> 6) \;\; i, j = 0,\ldots,3 \quad (8\text{-}62)$$

where $R_{ij}^{(m)}$ is specified in Equation 8-40, and where A is specified as:

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

$$A_{ij} = \begin{cases} 16 & , \quad (i,j) = \{(0,0),(0,2),(2,0),(2,2)\}; \\ 25 & , \quad (i,j) = \{(1,1),(1,3),(3,1),(3,3)\}; \\ 20 & , \quad otherwise. \end{cases} \tag{8-63}$$

For luma, QP = QP_Y, as specified in Equation 7-8 and 7-12.

The coefficients $Q_{ij}^{(m)}$, used in the formulas below, are specified as:

$$Q_{ij}^{(m)} = \begin{cases} M_{m0} & , \quad (i,j) = \{(0,0),(0,2),(2,0),(2,2)\}; \\ M_{m1} & , \quad (i,j) = \{(1,1),(1,3),(3,1),(3,3)\}; \\ M_{m2} & , \quad otherwise. \end{cases} \tag{8-64}$$

where the first and second subscripts of M are row and column indices, respectively, of the matrix specified as:

$$M = \begin{bmatrix} 13107 & 5243 & 8066 \\ 11916 & 4660 & 7490 \\ 10082 & 4194 & 6554 \\ 9362 & 3647 & 5825 \\ 8192 & 3355 & 5243 \\ 7282 & 2893 & 4559 \end{bmatrix} \tag{8-65}$$

The resulting sum, c^{PRIOR}, is quantised with a quantisation parameter QS, as given by:

$$c_{ij}^{QREC} = (\mathrm{Sign}(c_{ij}^{PRIOR}) * [\mathrm{Abs}(c_{ij}^{PRIOR})\, Q_{ij}^{(QS\%6)} + (1 << (14+QS/6))]) >> (15+QS/6) \quad i,j = 0,\ldots,3 \tag{8-66}$$

For luma, QS = QS_Y, which is specified in Equation 7-9.

These quantised levels, c^{QREC}, are scaled as given by:

$$w_{ij}^{QREC} = (c_{ij}^{QREC}\ R_{ij}^{(QS\%6)}) << (QS/6) \quad i,j = 0,\ldots,3 \tag{8-67}$$

where $R_{ij}^{(m)}$ is specified in Equation 8-40.

The transform and reconstruction processes are performed for these scaled levels, as specified in Equations 8-49 through 8-59. Finally, after the reconstruction of a macroblock, filtering takes place as described in subclause 8.7.

8.6.1.2 Chroma transform coefficient decoding process

Input to this process are inter predicted chroma samples for the current macroblock from subclause 8.4 and the prediction residual coefficient levels, ChromaDCLevel and ChromaACLevel.

Output of this process are the decoded chroma samples of the current macroblock prior to the deblocking filter process.

The decoding of chroma components for non -intra coded macroblocks in SP slices is similar to the decoding of luma components in SP slices.

The predicted chroma block, P_{ij}, where $P_{ij} = \{p_{00} \ldots p_{33}\}$, is transformed according to Equation 8-61 to produce transform coefficients c^{PRED}. After calculating the coefficients of the 4 chroma blocks of a macroblock, D C coefficients of the 4 chroma blocks are assembled into a 2x2 matrix of elements DC_{ij} and an additional 2x2 transform is applied to the DC coefficients of these blocks. The 2 dimensional 2x2 transform procedure is specified by:

$$c^{PRED} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} DC_{00} & DC_{01} \\ DC_{10} & DC_{11} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{8-68}$$

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

The received DC prediction residual coefficients, w^{QERR}, are scaled using quantisation parameter QP, and added to the DC transform coefficients of the prediction block, as given by:

$$c_{ij}^{PRIOR} = c_{ij}^{PRED} + (((w_{ij}^{QERR} \quad R_{00}^{(QP\%6)} \quad A_{00}) << (QP/6)) >>5) \quad i, j = 0,1 \tag{8-69}$$

The resulting sum, c^{PRIOR}, is quantised with a quantisation parameter QS, as given by:

$$c_{ij}^{QREC} = \{Sign(c_{ij}^{PRIOR}) \quad [Abs(c_{ij}^{PRIOR}) \quad Q_{00}^{(QS\%6)} + (1<<(15+QS/6))]\}>>(16+QS/6) \quad i, j = 0, 1 \tag{8-70}$$

2x2 transform process as defined in Equation 8-45 is applied to these quantized levels where the output of the transform is denoted by F_{ij}. After the transform, scaling is performed according to the following:

a) If QS is greater than or equal to 6, then the scaling result shall be calculated as

$$DC_{ij}^{QREC} = (F_{ij} * R_{00}^{(QS\%6)}) << (QS/6-1) \quad i, j = 0,1 \tag{8-71}$$

b) If QS is less than 6, then the scaling results shall be calculated by

$$DC_{ij}^{QREC} = (F_{ij} * R_{00}^{(QS\%6)}) >> 1 \quad i, j = 0,1 \tag{8-72}$$

AC coefficients are decoded in an identical process to that u sed for luma, as specified by Equations 8-62 to 8-67.

The transform and reconstruction processes are performed for the scaled levels, as specified in Equations 8-49 through 8-59. Finally, after the reconstruction of a macroblock, filtering takes place as described in subclause 8.7.

The values of QP and QS, denoted QP_C and QS_C, respectively, shall be obtained using the process descri bed in subclause 8.5.2.

8.6.2 SP and SI slice decoding process for switching pictures

This process is invoked, when decoding P macroblock types in SP slices in which sp_for_switch_flag is equal to 1 or SI macroblock type in SI slices.

Input to this process are the prediction residual transform levels and the predicted samples for the current macroblock.

Output of this process are the decoded samples of the current macroblock prior to deblocking.

This subclause applies to P macroblock t ypes in SP slices in which sp_for_switch_flag is equal to 1, and to SI macroblock type in SI slices.

Figure 8-20 illustrates the decoding for macroblocks in SI and SP slices that are not of the Intra type (Intra_4x4 or Intra_16x16). When decoding P macroblock types in SP slices, the prediction P_{ij} for the current coded macroblock of the slice being decoded is formed by the inter prediction block using the process specified in subclause 8.4. When decoding SI macroblock type in SI slices, the prediction P_{ij} is formed by intra prediction block using the process as specified in subclause 8.3.1. After forming the predicted block P_{ij}, the decoding of SI macroblock type in SI slices an d P macroblock types in SP slices follows the same steps.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

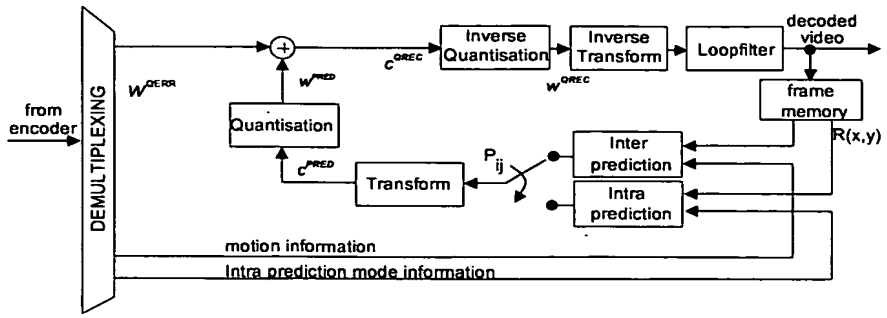


Figure 8-20– A block diagram of a conceptual decoder for non-intra macroblocks in SI slices; and for non-intra coded macroblocks in SP slices in which sp_for_switch_flag is equal to 1. (informative)

When decoding SI macroblocks, the intra prediction modes of the neighbouring SI and Intra_4x4 blocks are taken into account as described in subclause 8.3.1.

When decoding I_4x4 macroblocks, the intra prediction modes of the neighbouring Intra_4x4 blocks are taken into account as described in subclause 8.3.1, but the prediction mode of the neighbouring SI blocks are considered to be "mode 2: DC prediction", except in the case of constrained intra when the adjacent block is considered to be outside of the slice. [Ed.Note: read subclause 8.3.1 and be more specific here.]

8.6.2.1 Luma transform coefficient decoding process

Input to this process are predicted luma samples for the current macroblock from subclause 8.4 and the prediction residual coefficient levels, LumaLevel.

Output of this process are the decoded luma samples of the cur rent macroblock prior to the deblocking filter process.

The predicted luma block, P_{ij}, where $P_{ij} = \{p_{00} \ldots p_{33}\}$, is transformed according to Equation 8-61 to produce transform coefficients c^{PRED}. These transform coefficients are then quantised with a quantisation parameter QS, as given by:

$$w_{ij}^{PRED} = \{\text{Sign}(c_{ij}^{PRED})\,[\text{Abs}(c_{ij}^{PRED})\ Q_{ij}^{(QS\%6)} + (1<<(14+QS/6))]\}>>(15+QS/6) \qquad i, j = 0,\ldots, 3 \qquad (8\text{-}73)$$

NOTE - Equation 8-73 is the same as Equation 8-66 except for the change of name of input and output variables.

For luma, $QS = QS_Y$, which is specified in Equation 7-9.

The received prediction residual coefficients, w^{QERR}, are added to these quantised transform coefficients of the prediction block, as given by:

$$c_{ij}^{QREC} = w_{ij}^{PRED} + w_{ij}^{QERR} \qquad i, j = 0,\ldots, 3 \qquad (8\text{-}74)$$

These quantised levels, c^{QREC}, are scaled as in Equation 8-67 and the transform and reconstruction processes are performed for these scaled levels, as specified in Equations 8-49 through 8-59. Finally, after the reconstruction of a macroblock, filtering takes place as described in subclause 8.7.

8.6.2.2 Chroma transform coefficient decoding process

Input to this process are predicted chroma samples for the current macroblock from subclause 8.4 and the prediction residual coefficient levels, ChromaDCLevel and ChromaACLevel.

Output of this process are the decoded chroma samples of the current macroblock prior to the deblocking filter process.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

The decoding of chroma components for SP and SI non -intra macroblocks is similar to the decoding of luma components in SP slices.

The predicted chroma block, P_{ij}, where $P_{ij} = \{p_{00} \ldots p_{33}\}$, is transformed according to Equation 8-61 to produce transform coefficients c^{PRED}. After calculating the coefficients of the 4 chroma blocks of a macroblock, DC coefficients of the 4 chroma blocks are assembled into a 2x2 matrix of elements DC_{ij} and an additional 2x2 transform is applied to the DC coefficients as in Equation 8-68

The DC transform coefficients are then quantised with a quantisation parameter QS, as given by:

$$w_{ij}^{PRED} = \{\text{Sign}(c_{ij}^{PRED}) \ [\text{Abs}(c_{ij}^{PRED}) \ Q_{ij}^{(QS\%6)} + (1<<(15+QS/6))]\}>>(16+QS/6) \quad i, j = 0,1 \tag{8-75}$$

NOTE - Equation 8-75 is the same as Equation 8-70 except for the change of name of input and output variables.

The received prediction residual DC coefficients, w^{QERR}, are added to these quantised DC transform coefficients of the prediction block, as given by:

$$c_{ij}^{QREC} = w_{ij}^{PRED} + w_{ij}^{QERR} \qquad i, j = 0,1 \tag{8-76}$$

2x2 transform process as defined in Equation 8-45 is applied to these quantized levels c^{QREC} and scaling is performed according to Equation 8-71 and 8-72.

AC coefficients are decoded in an identical process to that used for luma, as specified in Equations 8-73 and 8-74.

The transform and reconstruction processes are performed for the scaled levels, as specified in Equations 8-49 through 8-59. Finally, after the reconstruction of a macroblock, filtering takes place as described in subclause 8.7.

The values of QP and QS, denoted QP_C and QS_C, respectively, shall be obtained using the process described in subclause 8.5.2.

8.7 Deblocking filter process

Input to this process are the decoded sample values of a picture, constructed as described in subclause 8.5.3; the prediction mode information (intra or inter) for each macroblock in a picture; informati on on which 4x4 luminance blocks in the picture were coded using non -zero coefficients; the motion vectors used for generating motion - compensated predictions in the picture; the quantization parameter values, QP, used for coding each macroblock; the deblocking filter control parameters disable_deblocking_filter_idc, FilterOffsetA, and FilterOffsetB; and, when mb_adaptive_frame_field_flag is equal to 1, the coding mode (frame/field) for each macroblock pair in a frame.

Output of this process are the filtered sample values for a picture that will be stored in the decoded picture buffer.

[Ed. Note: Operation with regard to PCM macroblock type needs to be specified.]

A conditional filtering shall be applied to all block edges of a slice, except edges at the bou ndary of the picture and any edges for which the deblocking filter is disabled as specified by disable_deblocking_filter_idc. This filtering is done on a macroblock basis, with macroblocks being processed in raster -scan order throughout the picture. For lu ma, as the first step, the 16 samples of the 4 vertical edges of the 4x4 raster shall be filtered beginning with the left edge, as shown on the left-hand side of Figure 8-21. Filtering of the 4 horizontal edges (vertical filtering) follows in the same manner, beginning with the top edge, as shown on the right -hand side of Figure 8-21. The same ordering applies for the filtering of each of the two chroma components, with the exception that 2 edges of 8 sample s each are filtered in each direction. [Ed. Note: The chroma filtering description here needs further clarification. Maybe we should do separate figures for chroma?] This process also affects the boundaries of the already reconstructed and filtered macrobl ocks above and to the left of the current macroblock. Picture edges are not filtered [Ed. Note: Is this redundant, based on the first sentence of the paragraph?].

For the deblocking filter process and the effect of the filter control parameters, the edges controlled by a macroblock are defined to be the block edges internal to the macroblock and the top and left edges between the current macroblock and its neighbouring macroblocks.

When mb_adaptive_frame_field_flag is equal to 1, all macroblock pairs are assumed to be reconstructed within the frame independent of the mb_field_decoding_flag.

When field_pic_flag is equal to 1, all decoding operations for the deblocking filter are based solely on samples within the current field.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

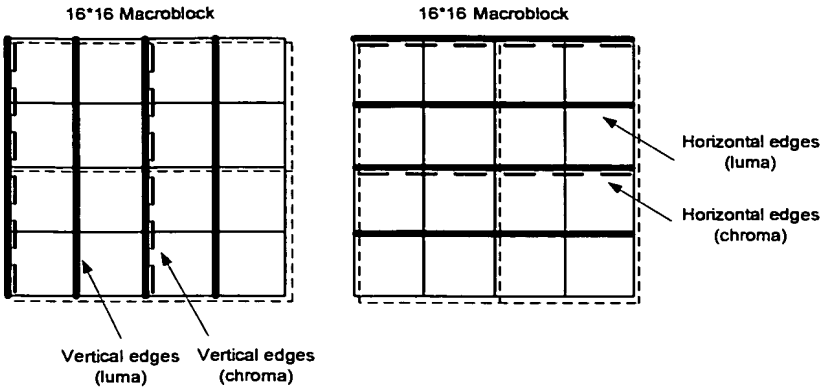


Figure 8-21 – Boundaries in a macroblock to be filtered (luma boundaries shown with solid lines and chroma boundaries shown with dashed lines)

Intra prediction of a macroblock shall be performed using the sample values prio r to the application of the deblocking filter process on the previously decoded and reconstructed neighbouring macroblocks.

NOTE - Depending on the implementation, the values necessary for intra prediction may need to be stored before filtering in order to be used in the intra prediction of the macroblocks to the right and below the current macroblock.

[Ed. Note: Does all of this really belong in the intra prediction section?]

8.7.1 Derivation process for the content dependent boundary filtering strength

Input to this process are the prediction mode information (intra or inter) for each macroblock in a picture; information on which 4x4 luminance blocks in the picture were coded using non -zero coefficients; the motion vectors used for generating motion-compensated predictions in the picture; and, when mb_adaptive_frame_field_flag is equal to 1 and field_pic_flag is equal to 0, the coding mode (frame/field) for each macroblock pair in a frame.

Output of this process are the Boundary Strength parameter, Bs, for each block boundary segment to be filtered.

For each boundary between neighbouring 4x4 luma blocks, a "Boundary Strength" Bs is assigned as shown in Figure 8-22. As indicated in Figure 8-21, every block boundary of a chroma block corresponds to a specific boundary of a luma block. Bs values for chroma are not calculated, but simply copied from the corresponding luma Bs. If Bs==0, filtering is skipped for that particular edge. In al l other cases filtering is dependent on the local sample properties and the value of Bs for this particular boundary segment.

For each edge, if any sample of the neighbouring macroblocks is coded using Intra macroblock prediction mode, selection between two relatively strong filtering modes is made based on whether the block boundary is also a macroblock boundary. If the block boundary is a macroblock boundary, a value of Bs=4 shall be assigned for this boundary segment. Otherwise, if Intra prediction is us ed in either block but the boundary is not a macroblock boundary, a value of Bs=3 shall be assigned.

If neither of the neighbouring macroblocks are coded using Intra macroblock prediction mode and at least one of them contains non-zero transform coefficient levels, a value of Bs=2 shall be assigned on this boundary segment, resulting in a medium filtering strength. If none of the previous conditions are satisfied, a value of Bs=1 shall be assigned if at least one of the following conditions is satisfied:

- one of the neigbouring macroblocks is coded in frame mode and the other is coded in field mode
- both macroblocks are coded in frame mode and the prediction of the two blocks is formed using different reference pictures or a different number [Ed.Note: ?] of re ference pictures
- both macroblocks are coded in field mode and the prediction of the same parity field of the two macroblocks is formed using different reference pictures or a different number of reference pictures.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

- any corresponding pair of motion vector s from the two neighbouring macroblocks is referencing the same picture and either component of this pair has a difference of more than one sample [Ed. Note: distance?].
- Otherwise, Bs shall be assigned the value 0, resulting in filtering being skipped fo r that particular boundary segment.

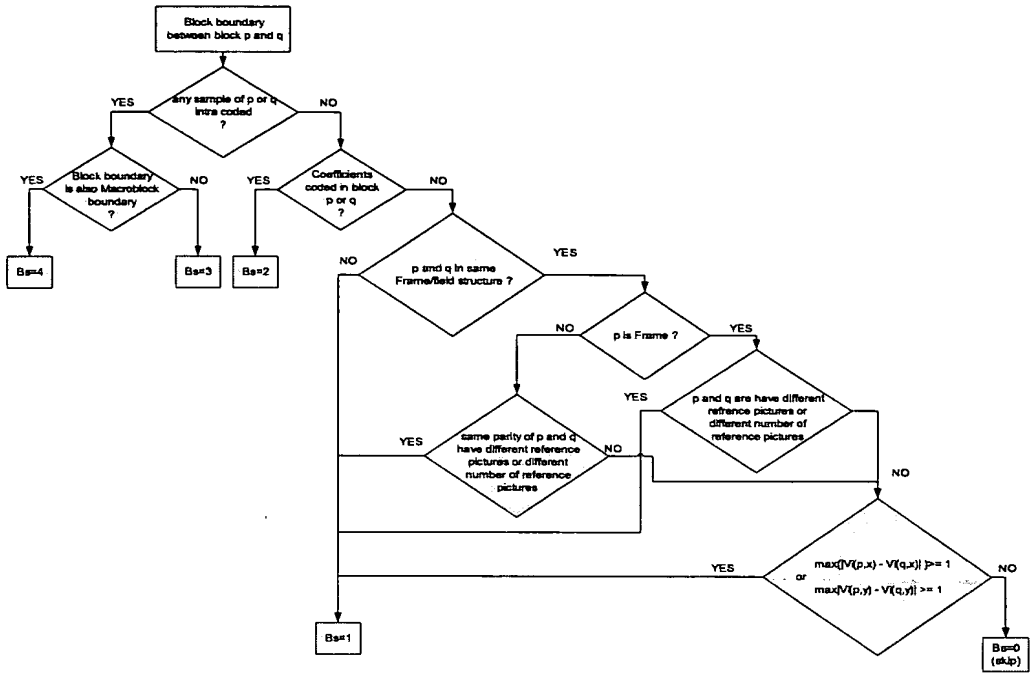


Figure 8-22 – Flow chart for determining the boundary strength (Bs), for the block boundary between two neighbouring 4x4 luma blocks [Ed.Note: ?] p and q , where $V_i(p,x)$ and $V_i(p, y)$ for i=1 to 1, 2 or 4, are the horizontal and vertical components of the motion vectors of block [Ed.Note: ?] p for the reference frame or the first and second reference fields.

8.7.2 Derivation process for the thresholds for each block boundary

Input to this process are the input sample values a single line of an edge segment to be filtered; the quantization parameter values, QP, used for coding each block that neighbours the edge segment; the deblocking filter control parameters FilterOffsetA and FilterOffsetB that apply to this edge segment; and the value of the Boundary Strength parameter, Bs, for the edge segment.

Output of this process is a Boolean decision as to wbether sample values will be modified on the input line of the edg e segment; and the value of Index $_A$, which is used to access the table of C0 values for filtering of edge segments with Bs smaller than 4, as specified in 8.7.3; the values of the threshold parameters (α and β), which are used for luma filtering of edge segments with Bs equal to 4, as specified in 8.7.4.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

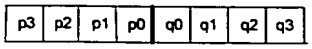


Figure 8-23 – Convention for describing samples across a 4x4 block horizontal or vertic al boundary

In the following description, the set of eight samples across a 4x4 block horizontal or vertical boundary is denoted as shown in Figure 8-23 with the actual boundary lying between p_0 and q_0. Uppercase letters indicate filtered samples and lower case letters indicate unfiltered samples with regard to the current edge filtering operation. However, p_1 and p_2 may indicate samples that have been modified by the filtering of a previous block edge.

Sets of samples across this edge are only filtered if the condition

$$Bs \neq 0 \;\&\&\; |p_0 - q_0| < \alpha \;\&\&\; |p_1 - p_0| < \beta \;\&\&\; |q_1 - q_0| < \beta \tag{8-77}$$

is true. The values of the thresholds α and β are dependent on the average value of QP for the two blocks as well as on a pair of index offsets "FilterOffsetA" and "FilterOffsetB" that may be transmitted in the slice header for the purpose of modifying the characteristics of the filter. The average QP value for the two blocks is computed as

$$QP_{av} = (QP_p + QP_q) >> 1. \tag{8-78}$$

For chroma, the corresponding QP_C values of the chroma blocks are used as input to to Equation 8-78 to compute QP_{av}.. When a macroblock pair is coded in field mode , the value used for QP_p or QP_q shall be the average of the QPs of the two macroblocks of the pair, which shall also be calculated using Equation 8-78, with QP_p and QP_q in this case being the QP values for the two macroblocks of the ma croblock pair. The index used to access the α table (Table 8-10), as well as the C0 table (Table 8-11) that is used in filtering of edges with Bs smaller than 4, is computed as:

$$Index_A = Clip3(0, 51, QP_{av} + FilterOffsetA) \tag{8-79}$$

NOTE - In SP and SI slices, QP_{av} is calculated in the same way as in other slice types. QS_Y from Equation 7-8 is not used in the deblocking filter.

The index used to access the β table (Table 8-10) is computed as:

$$Index_B = Clip3(0, 51, QP_{av} + FilterOffsetB) \tag{8-80}$$

The relationships between the indices computed using Equations 8-79 and 8-80 and the thresholds (α and β) are shown in Table 8-10.

Table 8-10 – QP_{av} and offset dependent threshold parameters α and β

	$Index_A$ (for α) or $Index_B$ (for β)																								
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
α	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	5	6	7	8	9	10	12	13
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	3	3	3	3	4	4	4

Table 8-10 (concluded)

	$Index_A$ (for α) or $Index_B$ (for β)																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
α	15	17	20	22	25	28	32	36	40	45	50	56	63	71	80	90	101	113	127	144	162	182	203	226	255	255
β	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

8.7.3 Filtering process for edges with Bs smaller than 4

Input to this process are the input sample values for a single line of an edge segment that is to be filtered; the value of the Boundary Strength parameter, Bs; and the value of Index $_A$, as specified in **Error! Reference source not found.**.

Output of this process are the filtered result sample values for each line of input sample values.

Two types of filtering are specified. In the case of edge segments with $0 < Bs < 4$, the samples having values p_0 and q_0 shall be modified to filtered values P_0 and Q_0 as given by:

$$\Delta = \text{Clip3}(-C, C, (((q_0 - p_0) << 2 + (p_1 - q_1) + 4) >> 3)) \quad (8\text{-}81)$$

$$P_0 = \text{Clip1}(p_0 + \Delta) \quad (8\text{-}82)$$

$$Q_0 = \text{Clip1}(q_0 - \Delta) \quad (8\text{-}83)$$

where C is determined as specified below.

The two intermediate threshold variables

$$a_p = | p_2 - p_0 | \quad (8\text{-}84)$$

$$a_q = | q_2 - q_0 | \quad (8\text{-}85)$$

shall be used to determine whether filtering for the luma samples p_1 and q_1 is taking place at this position of the edge.

If $a_p < \beta$ for a luma edge, the sample having value p_0 shall be modified to a filtered value P_1 as specified by

$$P_1 = p_1 + \text{Clip3}(-C0, C0, (p_2 + ((p_0 + q_0) >> 1) - (p_1 << 1)) >> 1) \quad (8\text{-}86)$$

If $a_q < \beta$ for a luma edge, the sample having value q_1 shall be modified to a filtered value Q_1 as specified by

$$Q_1 = q_1 + \text{Clip3}(-C0, C0, (q_2 + ((p_0 + q_0) >> 1) - (q_1 << 1)) >> 1) \quad (8\text{-}87)$$

where C0 is specified in Table 8-11. Chroma samples p_1 and q_1 are never filtered.

For luma, C is determined by setting it equal to C0 and then incrementing it by one if $a_p < \beta$, and again by one if $a_q < \beta$. For chroma, C is equal to C0 + 1.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 8-11 – Value of filter clipping parameter C0 as a function of $Index_A$ and Bs

	$Index_A$																								
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Bs = 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Bs = 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Bs = 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

Table 8-11 (concluded)

	$Index_A$																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Bs = 1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	4	4	4	5	6	6	7	8	9	10	11	13
Bs = 2	1	1	1	1	1	2	2	2	2	3	3	3	4	4	5	5	6	7	8	8	10	11	12	13	15	17
Bs = 3	1	2	2	2	2	3	3	3	4	4	4	5	6	6	7	8	9	10	11	13	14	16	18	20	23	25

8.7.4 Filtering process for edges for Bs equal to 4

Inputs to this process are the input sample values for a single line of an edge segment that is to be filtered; the values of the threshold parameters α and β for the edge segment, as specified in subclause 8.7.2.

Outputs of this process are the filtered result sample values for each line of input sample values.

In this subclause, the intermediate variables a_p and a_q are computed as specified in Equations 8-84 and 8-85.

For luma samples, if the following condition holds:

$$a_p < \beta \;\&\&\; |p_0 - q_0| < ((\alpha >> 2) + 2) \tag{8-88}$$

filtering of the left/upper side of the block edge shall be applied by replacing the samples having values p_0, p_1, and p_2 with the filtered values P_0, P_1, and P_2 according to:

$$P_0 = (p_2 + 2*p_1 + 2*p_0 + 2*q_0 + q_1 + 4) >> 3 \tag{8-89}$$

$$P_1 = (p_2 + p_1 + p_0 + q_0 + 2) >> 2 \tag{8-90}$$

$$P_2 = (2*p_3 + 3*p_2 + p_1 + p_0 + q_0 + 4) >> 3 \tag{8-91}$$

For chroma samples, and for luma samples for which the condition described in Equation 8-88 does not hold, the filter shall be applied by replacing the sample having value p_0 with the filtered value P_0 according to:

$$P_0 = (2*p_1 + p_0 + q_1 + 2) >> 2 \tag{8-92}$$

Similarly, for filtering of luma samples on the right/lower side of the edge, if the following condition holds:

$$a_q < \beta \;\&\&\; |p_0 - q_0| < ((\alpha >> 2) + 2) \tag{8-93}$$

filtering shall be applied by replacing the samples having values q_0, q_1, and q_2 with the filtered values Q_0, Q_1, and Q_2 according to:

$$Q_0 = (p_1 + 2*p_0 + 2*q_0 + 2*q_1 + q_2 + 4) >> 3 \tag{8-94}$$

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

$$Q_1 = (p_0 + q_0 + q_1 + q_2 + 2) >> 2 \quad (8\text{-}95)$$

$$Q_2 = (2*q_3 + 3*q_2 + q_1 + q_0 + p_0 + 4) >> 3 \quad (8\text{-}96)$$

For chroma samples, and for luma samples for which the condition described in Equation 8-93 does not hold, the filter shall be applied by replacing the sample having value q_0 with the filtered value Q_0 according to:

$$Q_0 = (2*q_1 + q_0 + p_1 + 2) >> 2 \quad (8\text{-}97)$$

9 Parsing process

Input to this process are bits from the RBSP.

Output of this process are syntax elements.

This process is invoked when the descriptor of a syntax element in the syntax tables in subclause 7.3 is equal to ue(v), se(v), te(v) (see subclause 9.1), ce(v) (see subclause 9.2), or ae(v) (see subclause 9.3).

9.1 Parsing process for exp-Golomb codes

This process is invoked when the descriptor of a syntax element in the syntax tables in subclause 7.3 is equal to ue(v), se(v), or te(v). For syntax elements in subclauses 7.3.4 and 7.3.5, this process is invoked only if entropy_coding_mode is equal to 0.

Input to this process are bits from the RBSP.

Output of this process are syntax elements.

This subclause specifies the parsing process for the exp-Golomb coded or truncated exp-Golomb coded syntax elements.

Syntax elements coded as ue(v) or se(v) are exp-Golomb coded. The parsing process for these syntax elements begins with reading the bits starting at the current location in the bitstream up to and including the first non-zero bit, and counting the number of leading bits that are equal to zero. This process shall be equivalent to the following:

```
LeadingZeroBits = -1;
for( B=0; !B; LeadingZeroBits++ )
    B = read_bits( 1 )
```

A code number is then assigned as follows:

$$CodeNum = 2^{LeadingZeroBits} - 1 + read\_bits(LeadingZeroBits)$$

where the value returned from read_bits(LeadingZeroBits) is interpreted as a binary representation of an unsigned integer with most significant bit written first.

CodeNum is used to determine syntax elements coded as ue(v) as specified in subclause 9.1.1 and se(v) as specified in subclause 9.1.2.

Table 9-1 illustrates the structure of the exp-Golomb code by separating the bit string into "prefix" and "suffix" bits. The "prefix" bits are those bits which are parsed in the above pseudo-code for the computation of LeadingZeroBits, and are shown as either 0 or 1 in the bit string column of Table 9-1. The "suffix" bits are those bits which are parsed in the computation of CodeNum and are shown as x_i in Table 9-1, with i being in the range 0 to LeadingZeroBits-1, inclusive. Each x_i can take on values 0 or 1.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-1 – Bit strings with "prefix" and "suffix" bits and assignment to CodeNum ranges

Bit string form	Range of CodeNum
1	0
0 1 x_0	1-2
0 0 1 x_1 x_0	3-6
0 0 0 1 x_2 x_1 x_0	7-14
0 0 0 0 1 x_3 x_2 x_1 x_0	15-30
0 0 0 0 0 1 x_4 x_3 x_2 x_1 x_0	31-62
...	...

Table 9-2 illustrates explicitly the assignment of bit strings to CodeNum values.

Table 9-2 – Code number and Exp-Golomb codewords in explicit form and used as ue(v)

Bit string	CodeNum
1	0
0 1 0	1
0 1 1	2
0 0 1 0 0	3
0 0 1 0 1	4
0 0 1 1 0	5
0 0 1 1 1	6
0 0 0 1 0 0 0	7
0 0 0 1 0 0 1	8
0 0 0 1 0 1 0	9
...	...

For syntax elements coded as te(v) the range of the syntax element shall be determined first. The range of this syntax element may be between 0 and X, with X being greater than or equal to 1.

If X is greater than 1, then CodeNum for syntax elements coded as te(v) shall be parsed in the same way as syntax elements coded as ue(v) or se(v).

If X is equal to 1, the parsing process for CodeNum for syntax elements coded as te(v) is given by a process equivalent to:

```
B = read_bits( 1 )
CodeNum = !B
```

CodeNum is used to determine the syntax elements coded as te(v) as specified in subclause 9.1.1.

9.1.1 Mapping process for unsigned exp-Golomb and for truncated unsigned exp-Golomb codes

Input to this process is CodeNum as specified in subclause 9.1.

Output of this process is a syntax element coded as ue(v) and te(v), which is equal to CodeNum.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

9.1.2 Mapping process for signed Exp-Golomb codes

Input to this process is CodeNum as specified in subclause 9.1.

Output of this process is the syntax element coded as se(v).

The syntax elements is assigned to the C odeNum by ordering the syntax element by its absolute value in increasing order and representing the positive value for a given absolute value with the lower CodeNum. Table 9-3 provides the assignment rule.

Table 9-3 – Assignment of syntax element to CodeNum for signed exp-Golomb coded syntax elements se(v)

CodeNum	syntax element
0	0
1	1
2	-1
3	2
4	-2
5	3
6	-3
k	$(-1)^{k+1}$ Ceil(k÷2)

9.2 CAVLC parsing process for transform coefficients

This process is invoked when the descriptor of a syntax element in the syntax tables in subclause 7.3.5.3.1 is equal to ce(v) and entropy_coding_mode is equal to 0.

Input to this process are bits from a slice layer RBSP.

Output of this process are the syntax elements with descriptor ce(v), which are coeff_token, trailing_ones_sign, coeff_level, total_zeros, and run_before.

The following syntax elements are used: [Ed.Note: move to semantics ?]

1. The syntax element coeff_token gi ves the total number of coefficients (from now referred to as TotalCoeffs) and also contains the number of trailing ones (from now referred to as T1s).

NOTE - If there are non-zero coefficients, it is typically observed that there is a string of coefficients at the highest frequencies that are ±1.

2. For T1s, the sign is derived from the syntax element trailing_ones_sign and the level magnitude is equal to 1.
3. For levels other than zero-valued levels or T1s, the syntax element coeff_level is used to derive thei r value.
4. For levels equal to 0, two syntax elements total_zeros and run_before are used to determine their position. The syntax element total_zeros is used to derive the total number number of zeros before all coefficients and the syntax element run_before is used to derive the number before each non-zero coefficient. Since the number of coefficients is known from the syntax element coeff_token, the range for the syntax elements total_zeros and run_before is limited given that a known maximum of levels that can be decoded for a block.

9.2.1 Parsing process for total number of coefficients and trailing ones

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

The syntax element coeff_token is decoded using the VLC specified in Table 9-4 to Table 9-7.

Four VLC tables are used for combined decoding of number of coefficients and T1s, i.e. one codeword signals both parameters. VLCs are listed in the tables below. T1s is clipped to 3. Any remaining trailing 1s are decoded as normal levels. The variable TotalCoeff is the value returned by the function total_coeff( ); the variable T1s is the value returned by the function tra iling_ones( ); and the variable NumCoeff is related to these quantities by TotalCoeff -T1s.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-4 – coeff_token mapping to total_coeff( ) and trailing_ones( ): Num -VLC0

total_coeff( )	trailing_ones( )			
	0	1	2	3
0	1	-	-	-
1	000101	01	-	-
2	00000111	000100	001	-
3	000000111	00000110	0000101	00011
4	0000000111	000000110	00000101	000011
5	00000000111	0000000110	000000101	0000100
6	0000000001111	00000000110	0000000101	00000100
7	0000000001011	0000000001110	00000000101	000000100
8	0000000001000	0000000001010	0000000001101	0000000100
9	00000000001111	00000000001110	0000000001001	00000000100
10	00000000001011	00000000001010	00000000001101	0000000001100
11	000000000001111	000000000001110	00000000001001	00000000001100
12	000000000001011	000000000001010	0000000000001101	00000000001000
13	0000000000001111	000000000000001	000000000001001	000000000001100
14	0000000000001011	0000000000001110	0000000000001101	000000000001000
15	0000000000000111	0000000000001010	0000000000001001	0000000000001100
16	0000000000000100	0000000000000110	0000000000000101	0000000000001000

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-5 – coeff_token mapping to total_coeff( ) and trailing_ones( ): Num -VLC1

total_coeff( )	trailing_ones( )			
	0	1	2	3
0	11	-	-	-
1	001011	10	-	-
2	000111	00111	011	-
3	0000111	001010	001001	0101
4	00000111	000110	000101	0100
5	00000100	0000110	0000101	00110
6	000000111	00000110	00000101	001000
7	00000001111	000000110	000000101	000100
8	00000001011	00000001110	00000001101	0000100
9	000000001111	00000001010	00000001001	000000100
10	000000001011	000000001110	000000001101	00000001100
11	000000001000	000000001010	000000001001	00000001000
12	0000000001111	0000000001110	0000000001101	000000001100
13	0000000001011	0000000001010	0000000001001	0000000001100
14	0000000000111	00000000001011	0000000000110	0000000001000
15	00000000001001	00000000001000	00000000001010	0000000000001
16	00000000000111	00000000000110	00000000000101	00000000000100

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-6 – coeff_token mapping to total_coeff( ) and trailing_ones( ): Num -VLC2

total_coeff( )	trailing_ones( )			
	0	1	2	3
0	1111	-	-	-
1	001111	1110	-	-
2	001011	01111	1101	-
3	001000	01100	01110	1100
4	0001111	01010	01011	1011
5	0001011	01000	01001	1010
6	0001001	001110	001101	1001
7	0001000	001010	001001	1000
8	00001111	0001110	0001101	01101
9	00001011	00001110	0001010	001100
10	000001111	00001010	00001101	0001100
11	000001011	000001110	00001001	00001100
12	000001000	000001010	000001101	00001000
13	0000001101	000000111	000001001	000001100
14	0000001001	0000001100	0000001011	0000001010
15	0000000101	0000001000	0000000111	0000000110
16	0000000001	0000000100	0000000011	0000000010

Table 9-7 – coeff_token mapping to total_coeff( ) and trailing_ones( ): Num -VLC_Chroma_DC

total_coeff( )	trailing_ones( )			
	0	1	2	3
0	01	-	-	-
1	000111	1	-	-
2	000100	000110	001	-
3	000011	0000011	0000010	000101
4	000010	00000011	00000010	0000000

9.2.1.1 Table selection process

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

For all elements, except chroma DC, a choice between three tables and one FLC is made. N is a value used for table selection. Selection is done as follows: N is calculated using total_coeff( ) of the block above and to the left of the current block: N_U and N_L. In the Table 9-8, X means that the block is available in the same slice. The block's coding mode is not taken into account when determining availability. When finding the block above and to the left for a block of Intra16x16 DC coefficients, the location of the block is assumed to be (0,0), i.e. the upper left corner of the macroblock.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-8 – Calculation of N for Num-VLCN

Upper block (N_U)	Left block (N_L)	N
X	X	$(N_L+N_U)/2$
X		N_U
	X	N_L
		0

0 <= N < 2 : Num-VLC0

2 <= N < 4 : Num-VLC1

4 <= N < 8 : Num-VLC2

N >= 8 : 6 bit FLC xxxxyy, as follows:

As a part of the coeff_token, TotalCoeff−1 is transmitted in the first 4 bits (xxxx) and T1s is transmitted as the last 2 bits (yy). There is one exception: the codeword 000011 represents T otalCoeff=0.

For chroma DC, Num-VLC_Chroma_DC is used.

9.2.2 Parsing process for level information

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

First, the sign of T1s are decoded from 1 bit each of trailing_ones_sign. A value of zero indicates a positive level and a value of 1 indicates a negative level. A maximum of 3 bits are read.

For the remaining level information, seven structured VLCs are used to decode levels. The structured level tables are explained in Table 9-9 to Table 9-15. Lev-VLC0 has its own structure while the other tables, **Lev-VLC*N*, *N* = 1 to 6**, share a common structure.

Table 9-9 – Level tables

Code	level_code
1	1
01	-1
001	2
0001	-2
..	..
00000000000001	-7
0000000000000001xxxs	±8 to ±15
0000000000000000001xxxxxxxxxxs	±16 ->

In the following descriptions of the level parsing process, Leadin gZeroBits specifies the number of zero-bits that occur before the first one-bit in the VLC codeword.

For Lev-VLC0, the following structure is used:

if (LeadingZeroBits < 14),

level_code is obtained directly from Table 9-9

else if (LeadingZeroBits < 15),

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

19-bit escape code: 0000 0000 0000 001x xxs,

where |level_code| = (value of x's) + 8,

and level_code is positive if s==0 and negative if s==1.

else,

28-bit escape code: 0000 0000 0000 0001 xxxx xxxx xxxs,

where |level_code| = (value of x's) + 16,

and level_code is positive if s==0 and negative if s==1.

For Lev-VLC*N*, *N* = 1 to 6, the following structure is used:

if (LeadingZeroBits < 15)

Code: 0....01x..xs,

where number of x's = *N*-1,

else,

28-bit escape code: 0000 0000 0000 0001 xxxx xxxx xxxs,

In either of these two cases:

|level_code| = (value of x's) + (LeadingZeroBits * $2^{(N-1)}$) + 1,

and level_code is positive if s==0 and negative if s==1.

Table 9-10 – Level VLC1

Code	level_code
1s	±1
01s	±2
..	..
0000000000000001s	±15
0000000000000001xxxxxxxxxxxs	±16 ->

Table 9-11 – Level VLC2

Code	level_code
1xs	±1 to ±2
01xs	±3 to ±4
..s	..
0000000000000001xs	±29 to ±30
0000000000000001xxxxxxxxxxxs	±31 ->

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-12 – Level VLC3

Code	level_code
1xxs	±1 to ±4
01xxs	±5 to ±8
..	..
0000000000000001xxs	±57 to ±60
00000000000000001xxxxxxxxxxxs	±61 ->

Table 9-13 – Level VLC4

Code	level_code
1xxxs	±1 to ±8
01xxxs	±9 to ±16
..	..
0000000000000001xxxs	±113 to ±120
00000000000000001xxxxxxxxxxxs	±121 ->

Table 9-14 – Level VLC5

Code	level_code
1xxxxs	±1 to ±16
01xxxxs	±17 to ±32
..	..
0000000000000001xxxxs	±225 to ±240
00000000000000001xxxxxxxxxxxs	±241 ->

Table 9-15 – Level VLC6

Code	level_code
1xxxxxs	±1 to ±32
01xxxxxs	±33 to ±64
..	..
0000000000000001xxxxxs	±449 to ±480
00000000000000001xxxxxxxxxxxs	±481 ->

When T1s is less than 3, the level of the first coefficient (after T1s) is equal to the decoded level_code plus 1:

```
if( first_coefficient && trailing_ones( ) < 3 )
    coeff_level = ( | level_code | + 1 ) * Sign( level_code )
```

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

```
else
    coeff_level = level_code
```

Otherwise, coefficient levels (coeff_level) are equal to the decoded level_code value given in Table 9-9 to Table 9-15.

The last two entries in the Lev-VLC0 table are escape codes. The first escape code with 19 bits, four "x"'s, is used to decode the 8 levels above the last regularly coded level. The next escape code with 28 -bits, 12 "x"'s, is used to decode all remaining levels. For Lev-VLC1 to Lev-VLC6 tables, only the 28-bit escape code is used.

9.2.2.1 Table selection

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

Selections of the tables are changed during the decoding process based on number of coefficients, number of trailing ones, and the previously decoded level value (coeff_level).

Let VLC denote the **Lev-VLC*N* (N=0-6) to be used.** After each level is decoded, the VLCN is updated according to the following method, where Level is the absolute value of the previously decoded level (coeff_level).

```
// VLC initialization for decoding first level
if (total_coeff(coeff_token) > 10 && trailing_ones(coeff_token)< 3)
   VLC = 1        // use Lev-VLC1 for first level
else
   VLC = 0        // use Lev-VLC0 for first level
// Assign FirstCoeff
// Decode level_code here and assign coeff_level and Level
// VLC update after decoding each level

vlc_inc table[6] = {0, 3, 6, 12, 24, 48}
if ((Level > vlc_inc[VLC]) && (VLC < 6))
   VLC++
if (FirstCoeff and Level > 3)
   VLC = 2
```

The first coefficient is always decoded with Lev-VLC0 or Lev-VLC1 while the rest of the coefficients are always decoded with Lev-VLC1 to Lev-VLC6.

The same procedure is used for chroma AC and DC coefficient levels.

9.2.3 Parsing process for run information

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

Run decoding is separated in total number of Zeros (i.e. the number of zeros located before the last non -zero coefficient in the zig-zag scan) and Run (of zeros) before each coefficient.

9.2.3.1 Parsing process for total number of zeros

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

The variable TotalZeros as given by total_zeros , is the sum of all zeros located before the last non -zero coefficient in a zig-zag scan. For example, given the string of coefficients 0 0 3 0 0 4 0 0 0 0 2 0 1 0 0 0, TotalZeros will be 2+2+4+1=9. Since TotalCoeff is already known, it determines the maximum possible value of TotalZeros. One out of 15 VLC tables is chosen based on TotalCoeff.

If TotalCoeff indicates that all coefficients are non-zero, TotalZeros is not decoded since it is known to be zero

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-16 – total_zeros tables for all 4x4 blocks

TotalCoeff TotalZeros	1	2	3	4	5	6	7	
0	1	111	0101	00011	0101	000001	000001	
1	011	110	111	111	0100	00001	00001	
2	010	101	110	0101	0011	111	101	
3	0011	100	101	0100	111	110	100	
4	0010	011	0100	110	110	101	011	
5	00011	0101	0011	101	101	100	11	
6	00010	0100	100	100	100	011	010	
7	000011	0011	011	0011	011	010	0001	
8	000010	0010	0010	011	0010	0001	001	
9	0000011	00011	00011	0010	00001	001	000000	
10	0000010	00010	00010	00010	0001	000000	-	
11	00000011	000011	000001	00001	00000	-	-	
12	00000010	000010	00001	00000	-	-	-	
13	000000011	000001	000000	-	-	-	-	
14	000000010	000000	-	-	-	-	-	
15	000000001	-	-	-	-	-	-	
TotalCoeff TotalZeros	8	9	10	11	12	13	14	15
0	000001	000001	00001	0000	0000	000	00	0
1	0001	000000	00000	0001	0001	001	01	1
2	00001	0001	001	001	01	1	1	-
3	011	11	11	010	1	01	-	-
4	11	10	10	1	001	-	-	-
5	10	001	01	011	-	-	-	-
6	010	01	0001	-	-	-	-	-
7	001	00001	-	-	-	-	-	-
8	000000	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-17 – TotalZeros table for chroma DC 2x2 blocks

NumCoeff TotalZeros	1	2	3
0	1	1	1
1	01	01	0
2	001	00	-
3	000	-	-

9.2.3.2 Parsing process for run before each coefficient

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

At this stage it is known how many zeros are left to distribute (call this ZerosLeft). When decoding a run_before for the first time, ZerosLeft begins at TotalZeros, and decreases as more run_before elements are decoded.

For example, if there is only 1 zero left, the run before the next coefficient must be either of length 0 or 1, and only one bit is needed.

The number of preceding zeros before each non -zero coefficient (called RunBefore) needs to be decoded to properly locate that coefficient. Before decoding the next RunBefore, ZerosLeft is updated and used to select one out of 7 tables. RunBefore does not need to be decoded in the following two situations:

- If the total number of zeros has been reached (ZerosLeft = 0)
- For the last coefficient in the reverse zig-zag scan. Then the value is known to be ZerosLeft. This also means that the maximum value to be decoded is 14.

Table 9-18 – Tables for run_before

ZerosLeft Run Before	1	2	3	4	5	6	>6
0	1	1	11	11	11	11	111
1	0	01	10	10	10	000	110
2	-	00	01	01	011	001	101
3	-	-	00	001	010	011	100
4	-	-	-	000	001	010	011
5	-	-	-	-	000	101	010
6	-	-	-	-	-	100	001
7	-	-	-	-	-	-	0001
8		-	-	-	-	-	00001
9	-	-	-	-	-	-	000001
10	-	-	-	-	-	-	0000001
11	-	-	-	-	-	-	00000001
12	-	-	-	-	-	-	000000001
13	-	-	-	-	-	-	0000000001
14	-	-	-	-	-	-	00000000001

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

9.3 CABAC parsing process for slice data

This process is invoked when parsing syntax elements with descriptor ae(v) in subclauses 7.3.4 and 7.3.5 and entropy_coding_mode is equal to 1.

Input to this process are bits from a slice layer RBSP and a request for a value of a syntax element .

Output of this process is the value of the syntax element.

When starting the parsing of the slice data of a slice in subclause 7.3.4, the initialisation process of the CABAC parsing process is invoked as specified in subclause 9.3.1.

The parsing of syntax elements proceeds as follows:

For each syntax element a binarization is derived as described in subclause 9.3.2.

The binarization for the syntax element and the sequence of parsed bins determines the decoding process flow as described in subcaluse 9.3.3.

For each bin of the binarization of the syntax element a context index Ctxldx is derived as specified in subclause 9.3.3.1.

For each Ctxldx the arithmetic decoding process is invoked as specified in subclause 9.3.3.2.

9.3.1 Initialisation process

Input to this process is the quantisation parameter QP_Y.

Output of this process are initialised CABAC internal variables.

The processes in subclauses 9.3.1.1 to 9.3.1.3 are invoked when starting the parsing of the slice data of a slice in subclause 7.3.4 and entropy_coding_ mode is equal to 1.

9.3.1.1 Initialisation process for contexts

Input to this process is the quantisation parameter QP_Y.

Output of this process are the initialised CABAC context models indexed by Ctxldx and the context identifier.

Each context model is initialised with an initial state, which consists of a state number and the meaning of the most probable symbol (MPS) as further described in subclause 9.3.3.2.2. The actual initial state of a context model depends linearly on the (initial) quantisation parameter QP_Y of the given slice, such that for each context model a pair of table entries {m, n} is given, from which the initial state is computed in the following way

1. Compute PreState = ((m*(QP_Y −12))>>4) + n;
2. Limit PreState to the range of [0,103], i.e.,
   PreState = min (103, max(0, PreState));
3. Map PreState to {State, MPS} pair according to the following rule:
   if (PreState <= 51) then {State = 51 -PreState, MPS = 0} else {State = PreState-52, MPS = 1}

For syntax elements specifically used for P- and B-slices only, there are always three tables defined. The index for addressing the used table is given by the indicator cabac_init_idc.

Table 9-19 – Table 9-29 contain the initialisation parameters {m, n} related to the context identifiers of all syntax elements except the end-of-slice flag.

For Ctxldx = 253 associated to the end_of_slice_flag and the bin of mb_type indicating the PCM mode, the initial state is given by State = 63 and MPS = 0.

NOTE – State=63 represents a non-adapted statewith fixed probability.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-19 – Initialisation parameters for context identifiers CtxMbTypeI, CtxMbTypeSIPrefix, and CtxMbTypeSISuffix

Context identifier	Initialisation parameter	CtxIdx										
		0	1	2	3	4	5	6	7	8	9	10
CtxMbTypeI	m	7	8	-2	-9	-15	-3	-1	0	na	na	na
	n	26	36	62	67	73	37	50	51	na	na	na
CtxMbTypeSIPrefix	m	7	8	-2	na	na	na	na	na	na	na	na
	n	26	36	62	na	na	na	na	na	na	na	na
CtxMbTypeSISuffix	m	na	na	na	7	8	-2	-9	-15	-3	-1	0
	n	na	na	na	26	36	62	67	73	37	50	51

Table 9-20 – Initialisation parameters for context identifier CtxMbSkip, CtxMbTypeP, CtxMbTypePSuffix, and CtxMbTypeBSuffix

Context identifier		CtxMbSkip			CtxMbTypeP				CtxMbTypePSuffix, CtxMbTypeBSuffix		
Value of cabac_init_idc	Initialisation parameter	CtxIdx									
		0	1	2	3	4	5	6	9	10	11
0	m	10	9	5	4	0	-29	2	-9	-6	0
	n	48	15	4	5	38	80	50	50	49	49
1	m	15	8	4	4	0	-27	18	-16	-14	1
	n	24	-3	-3	23	36	80	29	72	61	49
2	m	14	6	4	4	-4	-33	10	-15	-7	0
	n	31	-2	-1	8	40	81	50	50	39	50

Table 9-21 – Initialisation parameters for context identifier CtxMbTypeB, CtxB8ModeB, and CtxB8ModeP

Context identifier		CtxMbTypeB						CtxB8ModeB				CtxB8ModeP		
Value of cabac_init_idc	Initialisation parameter	CtxIdx												
		3	4	5	6	7	8	12	13	14	15	12	13	14
0	m	0	5	1	-25	-7	1	-7	-8	6	7	5	-2	7
	n	72	82	93	83	66	56	69	61	44	42	47	57	58
1	m	4	0	0	-32	-11	-1	-20	-6	11	7	4	-2	3
	n	86	103	103	90	68	59	112	65	48	59	37	45	61
2	m	-2	1	1	-36	-14	0	-17	-10	4	2	7	-2	9
	n	92	100	100	90	68	56	95	64	44	50	43	54	50

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-22 – Initialisation parameters for context identifiers CtxAbsMvdHoriz and CtxAbsMvdVert

Context identifier		CtxAbsMvdHoriz							CtxAbsMvdVert						
Value of cabac_init_idc	Initialisation parameter	CtxIdx													
		16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	m	-2	-4	-5	7	9	0	1	-2	-4	-6	5	6	-1	1
	n	52	60	64	48	61	69	83	47	59	64	44	59	65	81
1	m	-11	-11	-14	10	15	6	2	-5	-6	-9	4	7	-2	-1
	n	62	70	74	59	62	67	92	55	63	73	52	64	71	92
2	m	-10	-7	-7	1	4	-3	-2	-4	-4	-4	-2	-4	-7	-4
	n	57	64	66	54	67	76	95	45	54	59	57	80	78	87

Table 9-23 – Initialisation parameters for context identifier CtxRefIdx and CtxMbField

Context identifier		CtxRefIdx						CtxMbField		
Value of cabac_init_idc	Initialisation parameter	CtxIdx								
		30	31	32	33	34	35	46	47	48
0	m	-5	-1	0	0	-1	0	0	0	0
	n	30	52	49	64	51	47	29	46	55
1	m	1	0	3	0	-4	0	0	0	0
	n	84	53	59	53	29	59	29	46	55
2	m	5	-8	-1	-7	-17	-1	0	0	0
	n	74	64	65	65	56	58	29	46	55

Table 9-24 – Initialisation parameters for context identifiers CtxMbQpDelta, CtxIpredChroma, CtxIpredLumaMpm, CtxIpredLumaRm

Context label	CtxMbQpDelta		Context label	CtxIpredChroma		Context label	CtxIpredLumaMpm	
	m	n		m	n		m	n
36	0	29	40	-4	60	44	-5	50
37	0	51	41	1	67		CtxIpredLumaRm	
38	0	51	42	1	70	45	-3	57
39	0	51	43	1	59			

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-25 – Initialisation parameters for context identifiers CtxCbpLuma and CtxCbpChroma

Context identifier	CtxIdx	I-slices		P- and B-slices					
				cabac_init_idc = 0		cabac_init_idc = 1		cabac_init_idc = 2	
		m	n	m	n	m	n	m	n
CtxCbpLuma	49	-3	74	-23	86	-32	99	-28	92
	50	-4	62	-22	62	-34	76	-26	57
	51	-4	69	-18	64	-15	63	-22	72
	52	-5	55	-18	42	-21	49	-16	36
CtxCbpChroma	53	-7	64	-15	47	-24	60	-21	51
	54	-1	62	-11	61	-18	74	-19	76
	55	-9	76	-8	57	-14	62	-11	57
	56	-4	75	-13	71	-17	81	-17	81
	57	-18	65	-10	42	-15	46	-9	51
	58	-9	66	-8	59	-12	69	-16	81
	59	-13	69	-1	54	-1	49	-3	59
	60	-7	73	-7	69	-11	74	-17	86

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-26 – Initialisation parameters for context identifier CtxCodedBlock

Value of CtxCat	CtxIdx	I-slices		P- and B-slices					
				cabac_init_idc = 0		cabac_init_idc = 1		cabac_init_idc = 2	
		m	n	m	n	m	n	m	n
0	61	-7	71	-10	74	-7	81	-18	80
	62	-11	82	-12	84	-10	88	-17	85
	63	-11	82	-12	84	-11	93	-17	85
	64	-5	75	-12	85	-7	86	-17	86
1	65	-2	34	-3	33	-7	42	-4	33
	66	0	48	-2	50	1	51	0	50
	67	-6	50	-5	49	-14	57	-8	45
	68	-5	66	-11	77	-13	82	-10	74
2	69	-1	53	-3	58	-7	65	-3	57
	70	-6	68	-10	71	-11	81	-10	78
	71	-6	64	-8	67	-15	75	-8	62
	72	-22	102	-17	85	-22	97	-17	89
3	73	0	53	0	54	-2	57	-2	53
	74	1	61	2	58	-1	67	0	62
	75	0	60	1	58	-1	61	1	54
	76	-10	88	0	65	-5	76	-1	67
4	77	-2	39	-1	40	-2	38	-2	39
	78	-1	61	-2	58	0	64	-2	67
	79	-3	54	-3	57	-5	52	0	50
	80	-11	80	-6	67	-7	73	-10	77

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-27 – Initialisation parameters for context identifier CtxSigCoeff

Value of CtxCat	CtxIdx	I-slices		P- and B-slices					
				cabac_init_idc = 0		cabac_init_idc = 1		cabac_init_idc = 2	
		m	n	m	n	m	n	m	n
0	81	-1	65	-4	71	-6	74	-1	64
	82	-5	56	-8	62	-15	68	-12	52
	83	-1	57	-4	63	-5	67	-2	61
	84	-1	50	-6	58	-4	61	-2	57
	85	0	45	-3	51	-9	58	-8	48
	86	0	47	-3	53	-15	59	-1	37
	87	-6	55	-8	61	-17	64	-7	45
	88	0	40	-5	49	-17	56	-3	32
	89	0	44	-5	51	-11	59	-8	47
	90	0	60	-3	65	3	64	3	65
	91	0	49	-5	58	-1	57	-3	56
	92	-1	49	-6	58	-6	56	-1	50
	93	0	50	-3	58	-4	58	0	51
	94	0	51	-5	60	-5	59	-3	54
	95	0	61	-3	68	-3	68	-1	63
1	96	-4	48	-7	58	-22	57	-5	38
	97	3	53	0	59	1	63	0	60
	98	1	48	-2	53	5	56	-1	60
	99	2	42	-2	53	-12	50	0	37
	100	3	42	0	51	-9	41	0	38
	101	-2	36	-1	41	-9	31	-3	31
	102	0	39	-3	52	-10	38	-1	35
	103	0	50	0	56	-12	50	0	41
	104	1	59	0	56	7	69	0	70
	105	0	46	-6	51	-7	59	-1	51
	106	0	60	0	63	6	56	2	57
	107	1	55	7	53	7	46	1	51
	108	-2	51	-1	55	-5	49	-6	55
	109	2	73	1	81	6	78	7	69
2	110	-6	73	8	46	7	51	5	46
	111	-10	63	1	44	-10	47	-3	43
	112	-5	68	3	47	3	54	1	48

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

	113	-9	67	-2	49	1	56	0	51
	114	-8	63	0	47	-4	47	-6	47
	115	-3	51	3	44	-3	29	-5	45
	116	-7	41	-6	40	-2	11	-14	42
	117	-7	52	0	46	-8	29	-11	46
	118	-6	62	1	49	-7	48	-5	48
	119	-7	70	0	49	5	73	5	57
	120	-13	67	-8	50	0	68	0	54
	121	-2	64	3	53	3	55	1	52
	122	-1	56	4	50	-1	34	-5	51
	123	-8	59	-5	53	-4	33	-8	52
	124	1	79	9	72	3	87	10	64
3	125	-5	73	-3	63	0	62	-7	62
	126	-6	64	-1	53	-6	53	-4	47
	127	-3	78	3	61	5	71	0	71
4	128	0	51	5	48	-1	38	0	34
	129	0	63	6	47	4	63	3	53
	130	-3	57	0	40	0	64	2	54
	131	0	50	3	45	2	43	4	38
	132	6	45	8	48	13	17	9	28
	133	0	34	-2	42	14	0	0	26
	134	4	41	5	47	13	14	6	29
	135	3	49	7	45	4	40	6	39
	136	0	61	6	46	-4	76	2	61
	137	0	49	2	38	-7	65	1	49
	138	0	55	2	52	-3	58	4	49
	139	12	51	15	54	26	30	17	40
	140	-4	52	-4	52	12	28	-5	51
	141	4	67	2	71	-12	88	13	62

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-28 – Initialisation parameters for context identifier CtxLastCoeff

Value of CtxCat	CtxIdx	I-slices		P- and B-slices					
				cabac_init_idc = 0		cabac_init_idc = 1		cabac_init_idc = 2	
		m	n	m	n	m	n	m	n
0	142	5	33	12	25	12	13	18	22
	143	5	24	10	19	9	11	11	20
	144	-1	42	3	36	9	22	8	37
	145	4	32	11	21	17	13	10	33
	146	2	31	11	17	14	12	7	29
	147	3	31	13	18	13	14	10	25
	148	4	40	12	28	14	22	9	34
	149	3	31	14	15	12	14	12	22
	150	4	30	15	15	14	15	9	28
	151	6	46	14	35	23	26	13	49
	152	5	40	16	22	19	24	9	40
	153	5	35	16	18	17	20	9	31
	154	7	42	18	25	18	28	9	39
	155	6	44	15	29	17	30	9	40
	156	4	56	14	42	14	45	14	47
1	157	5	40	14	27	18	22	16	26
	158	8	44	16	32	24	27	13	43
	159	8	44	21	24	26	26	14	43
	160	4	44	15	26	19	23	15	27
	161	6	46	18	28	28	18	18	29
	162	7	45	24	16	40	10	23	22
	163	9	43	21	23	32	14	18	24
	164	5	51	18	31	21	27	16	32
	165	11	54	20	34	24	49	17	52
	166	15	40	17	31	16	40	13	35
	167	10	52	15	43	17	43	13	40
	168	11	56	22	45	22	44	5	44
	169	14	48	14	50	11	47	7	45
	170	8	65	7	73	8	71	6	61
2	171	25	0	20	18	33	4	20	12
	172	23	-4	24	12	29	1	27	1
	173	24	3	22	17	29	13	23	10

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

	174	24	5	23	17	25	20	23	12
	175	23	4	24	19	26	17	27	7
	176	26	2	25	21	32	11	28	9
	177	25	8	25	23	40	5	31	9
	178	27	0	23	25	28	12	25	13
	179	26	8	25	27	22	25	24	14
	180	25	23	26	33	25	48	29	27
	181	23	22	25	34	20	54	25	30
	182	24	24	22	40	20	47	22	31
	183	30	25	26	45	24	43	22	36
	184	25	32	19	51	14	48	17	40
	185	14	59	13	73	6	85	17	59
3	186	16	26	5	56	14	40	13	44
	187	16	26	12	47	14	34	18	36
	188	13	44	12	55	18	52	19	54
4	189	19	15	12	32	17	21	22	12
	190	15	23	12	35	9	36	15	22
	191	19	27	16	36	17	39	22	26
	192	22	26	18	36	18	38	25	20
	193	24	26	20	39	20	35	27	21
	194	27	33	22	44	31	33	29	28
	195	24	29	17	45	24	30	28	23
	196	24	30	19	44	19	38	24	27
	197	11	51	8	56	3	72	12	55
	198	20	40	13	49	9	60	23	42
	199	18	44	14	55	8	59	24	41
	200	20	51	13	66	22	53	27	46
	201	17	51	9	65	14	55	22	49
	202	6	71	0	82	-9	93	17	68

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-29 – Initialisation parameters for context identifier CtxAbsLevel

Value of CtxCat	CtxIdx	I-slices		P- and B-slices					
				cabac_init_idc = 0		cabac_init_idc = 1		cabac_init_idc = 2	
		m	n	m	n	m	n	m	n
0	203	0	50	-6	60	-7	62	-3	54
	204	-2	31	-6	37	-9	41	-8	33
	205	0	31	-1	36	-2	38	-5	36
	206	-1	39	-1	40	0	40	-3	41
	207	0	42	-1	44	0	45	-3	45
	208	0	46	-4	50	-6	53	-1	43
	209	0	50	-4	56	-4	56	-2	52
	210	-1	54	-4	59	-5	61	-3	56
	211	-4	58	-9	66	-6	65	-5	62
	212	-7	63	-12	74	-15	78	-16	75
1	213	-9	57	-8	59	-10	59	-7	55
	214	-4	22	-1	21	-10	30	-2	17
	215	-1	28	-2	32	-9	39	-2	28
	216	-1	34	-3	39	-6	43	-1	35
	217	-1	39	-1	43	-4	47	2	39
	218	-5	37	-8	44	-15	47	-8	40
	219	-5	44	-9	54	-12	55	-11	53
	220	-11	56	-9	59	-15	63	-11	61
	221	-10	58	-6	61	-15	67	-15	66
	222	-8	62	-8	68	-11	71	-19	74
2	223	-8	64	-4	53	-8	58	-6	53
	224	-10	35	-2	23	-10	38	-5	28
	225	-8	44	-3	32	-9	45	-7	34
	226	-6	47	-2	36	-6	47	-7	37
	227	-4	51	-2	41	-5	50	-6	39
	228	-6	50	-9	45	-16	55	-13	49
	229	-6	59	-5	52	-10	59	-7	53
	230	-6	62	-6	57	-10	63	-8	57
	231	-8	67	-7	61	-11	67	-7	59
	232	-14	78	-7	67	-12	73	-7	64
3	233	-7	71	-3	62	-4	64	0	58
	234	-11	49	-6	22	-12	38	-14	34
	235	-7	50	-4	33	-5	43	-10	40

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

	236	-2	46	-1	40	-2	46	-2	37
	237	-4	44	-6	45	-8	50	-8	45
	238	-11	60	-6	47	-7	51	-9	51
	239	-10	68	-7	58	-8	62	-7	58
	240	-14	75	-12	66	-12	72	-11	65
	241	-17	82	-11	68	-18	79	-13	71
4	242	-6	57	-4	51	-6	56	-6	54
	243	3	24	5	18	1	29	-8	32
	244	0	37	0	29	-3	40	-9	39
	245	-3	43	0	33	-4	43	-3	37
	246	-2	46	2	37	-3	45	0	37
	247	0	-8	48	-5	39	-10	47	-11
	248	-8	48	-5	39	-10	47	-11	45
	249	-9	60	-4	51	-11	60	-10	56
	250	-11	64	-5	56	-13	65	-10	60
	251	-4	63	0	57	-7	62	-1	56

9.3.1.2 Initialisation process for the arithmetic decoding engine

Output of this process are the initialised decoding engine register CodeIntRange and CodeIntValOffset.

In the initialisation procedure of the decod er, as shown in Figure 9-1, CodeIntValOffset is first filled with nine bits of the compressed data using read_bits( ) and then CodeIntRange is set to 0x01FE.

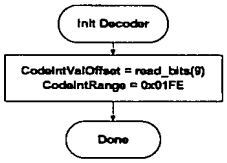


Figure 9-1 – Flowchart of initialisation of the decoding engine

9.3.1.3 Initialisation process for the arithmetic encoding engine (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

This process is invoked before encoding the first macroblock of a slice.

Output of this process are the values CodeIntLow and CodeIntRange of the arithmetic encoding engine.

In the initialisation procedure of the encoder, as illustrated in Figure 9-2, CodeIntLow is set to zero, and the range CodeIntRange is set to 0x01FE.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

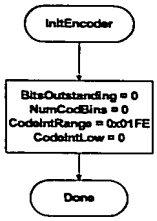


Figure 9-2 – Flowchart of initialisation of the encoding engine

9.3.2 Binarization process

Input to this process is a request for a value of a syntax element.

Output of this process is the binarization of the syntax element and the associated context identifier.

The binarization process is a mapping process of a possibly non-binary valued syntax element onto a set of binary representations of values of the syntax element. The binary representation consists of a sequence of binary decisions, so-called bins.

Table 9-30 specifies the type of binarization process and context identifier associated to each syntax element. The specification of the unary (U) binarization process, the truncated unary (TU) binarization process, the concatenated unary / k^{th}-order Exp-Golomb (UEGk) binarization process, and the fixed-length (FL) binarization process is given in subclauses 9.3.2.1 to 9.3.2.4.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-30 – Syntax elements and associated context identifiers

Syntax element	Context identifier	Type of binarization
mb_skip_flag	CtxMbSkip	FL, C_{max}=1
mb_type	CtxMbTypeI	as specified in subclause 9.3.2.5
	CtxMbTypeSIPrefix	as specified in subclause 9.3.2.5
	CtxMbTypeSISuffix	as specified in subclause 9.3.2.5
	CtxMbTypeP	as specified in subclause 9.3.2.5
	CtxMbTypeB	
	CtxMbTypePSuffix	as specified in subclause 9.3.2.5
	CtxMbTypeBSuffix	
	CtxB8ModeP	as specified in subclause 9.3.2.5
	CtxB8ModeB	
mvd_l0[0], mvd_l1[0]	CtxAbsMvdHoriz	as specified in subclause 9.3.2.6
mvd_l0[1], mvd_l1[1]	CtxAbsMvdVert	
ref_idx_l0, ref_idx_l1	CtxRefIdx	U
mb_qp_delta	CtxMbQpDelta	as specified in subclause 9.3.2.8
chroma_pred_mode	CtxIpredChroma	TU, C_{max}=3
prev_intra4x4_pred_mode_flag	CtxIpredLumaMpm	FL, C_{max}=1
rem_intra4x4_pred_mode	CtxIpredLumaRm	FL, C_{max}=7
mb_field_decoding_flag	CtxMbField	FL, C_{max}=7
coded_block_pattern	CtxCbpLuma	as specified in subclause 9.3.2.7
	CtxCbpChroma	
coded_block_flag	CtxCodedBlock	FL, C_{max}=1
significant_coeff_flag	CtxSigCoeff	FL, C_{max}=1
last_significant_coeff_flag	CtxLastCoeff	FL, C_{max}=1
coeff_absolute_value_minus1	CtxAbsLevel	UEG0, UCoff=14
coeff_sign_flag	CtxCoeffSignFlag	FL, C_{max}=1
end_of_slice_flag	CtxEosFlag	FL, C_{max}=1

9.3.2.1 Unary (U) binarization process

Input to this process is a request for a U binarization for a syntax element.

Output of this process is the U binarization of the syntax element.

Table 9-31 illustrates the unary binarization for a syntax element.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-31 – Unary binarization (informative)

syntax element	Binarization					
0	0					
1	1	0				
2	1	1	0			
3	1	1	1	0		
4	1	1	1	1	0	
5	1	1	1	1	1	0
...						
BinIdx	0	1	2	3	4	5

The binarization representation of a syntax element having value C is a bitstring of length C+1 indexed by BinIdx. The bins for BinIdx smaller than C are equal to 1. The bin with BinIdx equal to C is equal to 0.

9.3.2.2 Truncated unary (TU) binarization process

Input to this process is a request for a TU binarization for a syntax element and C_{max}.

Output of this process is the TU binarization of the syntax element .

For syntax element values smaller than C_{max} the unary binarization process as specified in subclause 9.3.2.1 is invoked. For the syntax element value equal to C_{max} the binary representation is a bitstring of length C_{max} +1 with all bins being equal to 1.

9.3.2.3 Concatenated unary/ k^{th}-order Exp-Golomb (UEGk) binarization process

Input to this process is a request for a UEGk binarization for a syntax element, parameter k and C_{max}.

Output of this process is the UEGk binarization of the syntax element.

Concatenated unary/k^{th}-order Exp-Golomb (UEGk) binarization consists of a concatenation of a prefix codeword and a suffix codeword. The prefix is formed by using a truncated unary binarization with C_{max} = UCoff, where UCoff denotes the cut-off parameter. For all syntax element values S with S < UCoff, the suffix codeword is void; for syntax element values S with S ≥ UCoff, the suffix consists of an Exp-Golomb code of order k for the symbol S – UCoff. For a given value S the Exp-Golomb code of order k is constructed as follows:

```
while( 1 ) {
    //first unary part of EGk
    if( S  >=  (unsigned int)( 1 << k ) )     {
       put( 1 );
       S = S - ( 1<<k );
       k++;
    } else {
       put( 0 );      //now terminating zero of unary part of EGk
       while( k-- )  //finally binary part of EGk
          put( ( S>>k )  &  0x01 );
       break;
    }
}
```

The indexing of bins is such that bin number BinIdx = 0 relates to the first binary decision in the unary prefix code with increasing indices towards the least significant bits of the binary part of the Exp-Golomb suffix.

9.3.2.4 Fixed-length (FL) binarization process

Input to this process is a request for a FL binarization for a syntax element and C_{max}.

Output of this process is the FL binarization of the syntax element.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Fixed-length (FL) binarization is constructed by using a L-bit binary representation of the syntax element value, where L = ceil($\log_2$(C_{max}+1)). The indexing of bins for the fixed-length binarization is such that the BinIdx = 0 relates to the least significant bit with increasing bin indices towards the most significant bit.

9.3.2.5 Binarization process for macroblock type and sub-macroblock type

Input to this process is a request for a binarization for syntax elements mb_type or sub_mb_type.

Output of this process is the binarization of the syntax element.

The binarization scheme for decoding of macroblock type in I slices is specified in Table 9-32. For macroblock types in SI slices the binarization consists of a prefix and a suffix part. The prefix consists of a single bit b_0 = ((mb_type == SI) ? 0 : 1). The suffix part for mb_type SI is void, while the suffix parts for all other macroblock types are given by the corresponding binarization pattern specified in Table 9-32.

The binarization schemes for P, SP, and B slices are specified in Table 9-33. The binarization for intra macroblock types in P and SP slices corresponding to mb_type values 7 to 32 consists of a prefix as specified in Table 9-33 and a suffix as specified in Table 9-32 for the corresponding intra mb_type. For intra macroblock types in B slices (mb_type values 23 to 48) the binarization consists of a prefix as specified in Table 9-33 and a suffix as specified in Table 9-32 for the corresponding intra mb_type.

For P, SP, and B slices the specification of the binarization for sub_mb_type is given in Table 9-34.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-32 – Binarization for macroblock types for I slices

Value (name) of mb_type	Binarization						
0 (I_4x4)	0						
1 (I_16x16_0_0_0)	1	0	0	0	0	0	
2 (I_16x16_1_0_0)	1	0	0	0	0	1	
3 (I_16x16_2_0_0)	1	0	0	0	1	0	
4 (I_16x16_3_0_0)	1	0	0	0	1	1	
5 (I_16x16_0_1_0)	1	0	0	1	0	0	0
6 (I_16x16_1_1_0)	1	0	0	1	0	0	1
7 (I_16x16_2_1_0)	1	0	0	1	0	1	0
8 (I_16x16_3_1_0)	1	0	0	1	0	1	1
9 (I_16x16_0_2_0)	1	0	0	1	1	0	0
10 (I_16x16_1_2_0)	1	0	0	1	1	0	1
11 (I_16x16_2_2_0)	1	0	0	1	1	1	0
12 (I_16x16_3_2_0)	1	0	0	1	1	1	1
13 (I_16x16_0_0_1)	1	0	1	0	0	0	
14 (I_16x16_1_0_1)	1	0	1	0	0	1	
15 (I_16x16_2_0_1)	1	0	1	0	1	0	
16 (I_16x16_3_0_1)	1	0	1	0	1	1	
17 (I_16x16_0_1_1)	1	0	1	1	0	0	0
18 (I_16x16_1_1_1)	1	0	1	1	0	0	1
19 (I_16x16_2_1_1)	1	0	1	1	0	1	0
20 (I_16x16_3_1_1)	1	0	1	1	0	1	1
21 (I_16x16_0_2_1)	1	0	1	1	1	0	0
22 (I_16x16_1_2_1)	1	0	1	1	1	0	1
23 (I_16x16_2_2_1)	1	0	1	1	1	1	0
24 (I_16x16_3_2_1)	1	0	1	1	1	1	1
25 (PCM)	1	1					
BinIdx	0	1	2	3	4	5	6

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-33 – Binarization for macroblock types for P, SP, and B slices

Slice type	Value (name) of mb_type	Binarization						
P, SP slice	0 (P_L0_16x16)	0	0	0				
	1 (P_L0_L0_16x8)	0	1	1				
	2 (P_L0_L0_8x16)	0	1	0				
	4 (P_8x8)	0	0	1				
	6 (P_8x8ref0)	na						
	7 to 32 (Intra, prefix only)	1						
B slice	0 (B_Direct_16x16)	0						
	1 (B_L0_16x16)	1	0	0				
	2 (B_L1_16x16)	1	0	1				
	3 (B_Bi_16x16)	1	1	0	0	0	0	
	4 (B_L0_L0_16x8)	1	1	0	0	0	1	
	5 (B_L0_L0_8x16)	1	1	0	0	1	0	
	6 (B_L1_L1_16x8)	1	1	0	0	1	1	
	7 (B_L1_L1_8x16)	1	1	0	1	0	0	
	8 (B_L0_L1_16x8)	1	1	0	1	0	1	
	9 (B_L0_L1_8x16)	1	1	0	1	1	0	
	10 (B_L1_L0_16x8)	1	1	0	1	1	1	
	11 (B_L1_L0_8x16)	1	1	1	1	1	0	
	12 (B_L0_Bi_16x8)	1	1	1	0	0	0	0
	13 (B_L0_Bi_8x16)	1	1	1	0	0	0	1
	14 (B_L1_Bi_16x8)	1	1	1	0	0	1	0
	15 (B_L1_Bi_8x16)	1	1	1	0	0	1	1
	16 (B_Bi_L0_16x8)	1	1	1	0	1	0	0
	17 (B_Bi_L0_8x16)	1	1	1	0	1	0	1
	18 (B_Bi_L1_16x8)	1	1	1	0	1	1	0
	19 (B_Bi_L1_8x16)	1	1	1	0	1	1	1
	20 (B_Bi_Bi_16x8)	1	1	1	1	0	0	0
	21 (B_Bi_Bi_8x16)	1	1	1	1	0	0	1
	22 (B_8x8)	1	1	1	1	1	1	
	23 to 48 (Intra, prefix only)	1	1	1	1	0	1	
BinIdx		0	1	2	3	4	5	6

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-34 – Binarization for sub-macroblock types in P, SP, and B slices

Slice type	Value (name) of sub_mb_type	Binarization					
P, SP slice	0 (P_L0_8x8)	1					
	1 (P_L0_8x4)	0	0				
	2 (P_L0_4x8)	0	1	1			
	3 (P_L0_4x4)	0	1	0			
B slice	0 (B_Direct_8x8)	0					
	1 (B_L0_8x8)	1	0	0			
	2 (B_L1_8x8)	1	0	1			
	3 (B_Bi_8x8)	1	1	0	0	0	
	4 (B_L0_8x4)	1	1	0	0	1	
	5 (B_L0_4x8)	1	1	0	1	0	
	6 (B_L1_8x4)	1	1	0	1	1	
	7 (B_L1_4x8)	1	1	1	0	0	0
	8 (B_Bi_8x4)	1	1	1	0	0	1
	9 (B_Bi_4x8)	1	1	1	0	1	0
	10 (B_L0_4x4)	1	1	1	0	1	1
	11 (B_L1_4x4)	1	1	1	1	0	
	12 (B_Bi_4x4)	1	1	1	1	1	
BinIdx		0	1	2	3	4	5

9.3.2.6 Binarization process for motion vectors

Input to this process is a request for a binarization for the syntax element mvd_l0 or mvd_l1.

Output of this process is the binarization of the syntax element.

The binarization of the motion vector difference mvd_l0[i][j] or mvd_l 1[i][j] consists of a prefix and a suffix part. The prefix part of the binarization is given by the concatenated unary/3rd-order Exp-Golomb (UEG3) binarization of the absolute value of mvd_lx[i][j] (x = 0,1) with cut-off parameter UCoff = 9. The suffix part consists of the FL binarization of ½*(1-Sign(mvd_lx[i][j])) (x = 0,1) with C_{max} = 1.

9.3.2.7 Binarization process for coded block pattern

Input to this process is a request for a binarization for the syntax element coded_block_pattern.

Output of this process is the binarization of the syntax element.

The binarization of coded block pattern consists of a prefix and a suffix part. The prefix part of the binarization is given by the FL binarization of CodedBlockPatternLuma with C_{max} = 15 and L = 4. The suffix part consists of the TU binarization of CodedBlockPatternChroma with C_{max} = 2. The relationship between the value of the syntax element coded_block_pattern and the values of CodedBlockPatternLuma and CodedBlockPatternChroma is given as specified in subclause 7.4.5: coded_block_pattern = CodedBlockPatternLuma + 16 × CodedBlockPatternChroma.

9.3.2.8 Binarization process for mb_qp_delta

Input to this process is a request for a binarization for the syntax element mb_qp_delta.

Output of this process is the binarization of the syntax element.

The binary representation of mb_qp_delta is obtained by the U binarization of the mapped value of the syntax element mb_qp_delta, where the assignment rule between the signed value of mb_qp_delta and its m apped value is given as specified in Table 9-3.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

9.3.3 Decoding process flow

Input to this process is a binarization of the syntax element to be parsed and its context identifier as specified in subclause 9.3.2.

Output of this process is the value of the syntax element.

This process specifies how each bit of a bitstring is parsed for each syntax element.

After parsing each bit, the resulting bitstring is compared to one of the binary representations of the bin arization of the syntax element.

If the bitstring is equal to one of the binary representations, the corresponding value of the syntax element is returned. Otherwise, the next bit is parsed as described below.

While parsing each bin of the binarization, t he variable BinIdx is incremented by one starting with BinIdx=0 for the first bin. The variable BinIdx and the context identifier are used to determine the CtxIdx as specified in subclause 9.3.3.1.

The CtxIdx is used to parse th e value of the bin from the bitstream as specified in subclause 9.3.3.2.

9.3.3.1 Derivation process for the CtxIdx

Input to this process is the BinIdx and the context identifier.

Output of this process is the CtxIdx.

Table 9-35 shows the assignment of MaxCtxBinIdx and the ranges of CtxIdx to context identifiers.

All bins with BinIdx greater than MaxCtxBinIdx are parsed using the context assigned to MaxCtxBinIdx. If in Table 9-35 an entry for MaxCtxBinIdx is labeled as "na", no bins with BinIdx greater than MaxCtxBinIdx exist.

For all bins with BinIdx greater or equal to MinBinIdxEqProb the variable CtxEqPropFlag is set to 1. For bins with BinIdx smaller than MinBinIdxEqProb or for all bi ns of context identifiers, where in Table 9-35 an entry for MaxBinIdxEqProb is labeled as "na", CtxEqPropFlag is set to 0. CtxEqPropFlag is used to indicate that for parsing the value of the bin from the bitstream the arithmetic de coding process for decisions with uniform pdf as specified in subclause 9.3.3.2.2.3 shall be applied.

The context indices CtxIdx are labeled in a range from 0 to 253. The assignments of context identifier to ranges of CtxIdx is specified in Table 9-35. The lower value of the range of CtxIdx assigned to a syntax element specifies the offset for the CtxIdx (CtxIdxOffset).

NOTE – Some syntax elements can never occur together in a slice and therefore sometimes share the same CtxIdx.

CtxIdx=253 is assigned to the syntax element end_of_slice_flag and the bin of mb_type indicating the PCM mode. It represents a context, where the probabilities are fixed and not adapted. For parsing the value of the corresponding b ins from the bitstream, the arithmetic decoding process for decisions before termination as specified in subclause 9.3.3.2.2.4 shall be applied.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-35 – Context identifiers and associated MaxBinCtxIdx and ranges of CtxIdx

Context identifier	MaxCtxBinIdx	MinBinIdxEqProb	Range of CtxIdx
CtxMbSkip	na	na	0 – 2
CtxMbTypeI	na	na	0 – 7
CtxMbTypeSIPrefix	na	na	0 – 2
CtxMbTypeSISuffix	na	na	3 – 10
CtxMbTypeP	na	na	3 – 6
CtxMbTypeB	3	na	3 – 8
CtxMbTypePSuffix	4	na	6 – 9
CtxMbTypeBSuffix	4	na	8 – 11
CtxB8ModeP	na	na	12 – 14
CtxB8ModeB	3	na	12 – 15
CtxAbsMvdHoriz	4	10	16 – 22
CtxAbsMvdVert	4	10	23 – 29
CtxRefIdx	2	na	30 – 35
CtxMbQpDelta	2	na	36 – 39
CtxIpredChroma	1	na	40 – 43
CtxIpredLumaMpm	na	na	44
CtxIpredLumaRm	0	na	45
CtxMbField	na	na	46 – 48
CtxCbpLuma	na	na	49 – 52
CtxCbpChroma	na	na	53 – 60
CtxCodedBlock	na	na	61 – 80
CtxSigCoeff	na	na	81 – 141
CtxLastCoeff	na	na	142– 202
CtxAbsLevel	1	15	203 – 252
CtxSignCoeffFlag	na	0	na
CtxEosFlag	na	na	253 (Uses Decode_terminate)

Table 9-36 shows the assignment of CtxIdx increments (CtxIdxInc) to BinIdx for all context identifiers except CtxCodedBlock, CtxSigCoeff, CtxLastCoeff, and CtxAbsLevel.

For all context identifiers except CtxCodedBlock, CtxSigCoeff, CtxLastCoeff, and CtxAbsLevel, the range of CtxIdx assigned to BinIdx is specified by CtxIdxOffset + CtxIdxInc.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-36 – Assignment of CtxIdxInc to BinIdx for all context identifiers except CtxCodedBlock, CtxSigCoeff, CtxLastCoeff, and CtxAbsLevel

Context identifier	BinIdx					
	0	1	2	3	4	≥ 5
CtxMbSkip	0,1,2	-/-	-/-	-/-	-/-	-/-
CtxMbTypeI	0,1,2	CtxIdx=253; (Uses Decode_terminate)	3	4	5,6	6,7
CtxMbTypeSIPrefix	0,1,2	-/-	-/-	-/-	-/-	-/-
CtxMbTypeSISuffix	0,1,2	CtxIdx=253; (Uses Decode_terminate)	3	4	5,6	6,7
CtxMbTypeP	0	1	2,3	-/-	-/-	-/-
CtxMbTypePSuffix	0	1	2	2,3	3	3
CtxMbTypeB	0,1,2	3	4,5	5	5	5
CtxMbTypeBSuffix	0	1	2	2,3	3	3
CtxB8ModeP	0	1	2	-/-	-/-	-/-
CtxB8ModeB	0	1	2,3	3	3	3
CtxAbsMvdHoriz	0,1,2	3	4	5	6	6
CtxAbsMvdVert	0,1,2	3	4	5	6	6
CtxRefIdx	0,1,2,3	4	5	5	5	5
CtxMbQpDelta	0,1	2	3	3	3	3
CtxIpredChroma	0,1,2	3	3	-/-	-/-	-/-
CtxIpredLumaRm	0	0	0			
CtxMbField	0,1,2	-/-	-/-	-/-	-/-	-/-
CtxCbpLuma	0,1,2,3	0,1,2,3	0,1,2,3	0,1,2,3	-/-	-/-
CtxCbpChroma	0,1,2,3	4,5,6,7	-/-	-/-	-/-	-/-
CtxSignCoeffFlag	0	-/-	-/-	-/-	-/-	-/-
CtxEosFlag	0	-/-	-/-	-/-	-/-	-/-

Table 9-37 shows the assignment of CtxIdx incremen ts (CtxIdxInc) to CtxCat for context identifiers CtxCodedBlock, CtxSigCoeff, CtxLastCoeff for BinIdx equal to 0. Depending on CtxCat, the range for the context identifiers CtxCodedBlock, CtxSigCoeff, CtxLastCoeff assigned to BinIdx is specified by CtxIdxOf fset + CtxIdxInc(CtxCat). Note that the specification of CtxCat is given in Table 9-41.

Table 9-37 – Assignment of CtxIdxInc to CtxCat for context identifiers CtxCodedBlock, CtxSigCoef f, and CtxLastCoeff

Context identifier	CtxCat (as specified in Table 9-41)				
	0	1	2	3	4
CtxCodedBlock	0 – 3	4 – 7	8 – 11	12 – 15	16 – 19
CtxSigCoeff	0 – 14	15 – 28	29 – 43	44 –46	47 – 60
CtxLastCoeff	0 – 14	15 – 28	29 – 43	44 –46	47 – 60

Table 9-38 shows the assignment of CtxIdx increments (CtxIdxInc) to BinIdx and CtxCat for context identifier CtxAhsLevel. Depending on CtxCat, the range for the context identifier CtxAbsLevel assigned to BinIdx is specified by CtxIdxOffset + CtxAbsLevelOffset(CtxCat) + CtxIdxInc(CtxCat).

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table 9-38 – Assignment of CtxAbsLevelOffset and CtxIdxInc to CtxCat for context identifier CtxAbsLevel

CtxCat	CtxAbsLevelOffset	BinIdx=0	BinIdx > 0
0	0	0,1,2,3,4	5,6,7,8,9
1	10	0,1,2,3,4	5,6,7,8,9
2	20	0,1,2,3,4	5,6,7,8,9
3	30	0,1,2,3,4	5,6,7,8
4	39	0,1,2,3,4	5,6,7,8,9

Further specification of CtxIdxInc for all context identifiers with the ranges of CtxIdx depending on BinIdx or CtxCat are given in subclauses 9.3.3.1.1 to 9.3.3.1.3.

For some BinIdx and context identifiers, CtxIdxInc is derived using context templates applied to prior in decoding order parsed syntax elements or parts thereof as specified in subclause 9.3.3.1.1.

For some BinIdx and context identifiers, CtxIdxInc is derived using context templates applied to prior in decoding order parsed bins as specified in subclause 9.3.3.1.2.

For the context identifiers CtxCodedBlock, CtxSigCoeff, CtxLastCoeff, and CtxAbsLevel, CtxIdxInc is derived as specified in subclause 9.3.3.1.3.

9.3.3.1.1 Assignment process of CtxIdxInc for context templates using two neighbouring syntax elements

Input to this process are a context identifier and a BinIdx.

Output of this process is a context index CtxIdxInc.

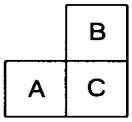


Figure 9-3 – Illustration of the generic context template using two neighbouring syntax elements A and B for conditional decoding of a current syntax element C

The generic design of this type of context template is shown in Figure 9-3. It involves two prior decoded syntax elements or prior decoded bins of the same syntax element that correspond to the spatially neighbouring blocks to the left (A) and on the top (B) of the regarded block C. The generic form of the equation specifying this type of context is given by

$$CtxVarSpat = CondTerm(A, B) \quad (9\text{-}1)$$

where the conditioning term CondTerm(A, B) describes the functional relationship between the spatially neigbbouring prior decoded syntax elements A and B, on the one hand, and CtxIdxInc, on the other hand. Three special cases of this template are specified as follows:

$$CtxVarSpat1 = CondTerm(A) + CondTerm(B) \quad (9\text{-}2)$$

$$CtxVarSpat2 = CondTerm(A) + 2*CondTerm(B) \quad (9\text{-}3)$$

$$CtxVarSpat3 = CondTerm(A). \quad (9\text{-}4)$$

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-39 – Specification of CondTerm for context identifiers and BinIdx using context templates according to Equations (9-2) – (9-4)

Context identifier	BinIdx	Context template	CondTerm(X), semantics of X	CondTerm(X), if X not available
CtxMbSkip	0	CtxVarSpat1	(mb_skip_flag(X) == 0) ? 1 : 0 X: neighbouring macroblock	0
CtxMbTypeI	0	CtxVarSpat1	(mb_type(X) != I_4x4) ? 1: 0 X: neighbouring macroblock	0
CtxMbTypeSIPrefix	0	CtxVarSpat1	(mb_type(X) != SI) ? 1: 0 X: neighbouring macroblock	0
CtxMbTypeSISuffix	0	CtxVarSpat1	(mb_type(X) != I_4x4) ? 1: 0 X: neighbouring macroblock	0
CtxMbTypeB	0	CtxVarSpat1	((mb_type(X) != B_Direct_16x16) ? 1: 0) X: neighbouring macroblock	0
CtxIpredChroma	0	CtxVarSpat1	((intra_chroma_pred_mode(X) != 0) ? 1: 0) X: neighbouring macroblock	0
CtxRefIdx	0	CtxVarSpat2	(ref_idx_l0 != 0 bzw. ref_idx_l1(X) != 0) ? 1: 0 X: neighbouring block	0
CtxMbField	0	CtxVarSpat1	((mb_field_decoding_flag(X) != 0) ? 1: 0) X: neighbouring macroblock pair	0
CtxCbpLuma	i=0-3	CtxVarSpat2	i-th bit of CodedBlockPatternLuma(X) X: neighbouring 8x8 block of i-th block	0
CtxCbpChroma	0	CtxVarSpat2	(CodedBlockPatternChroma(X) != 0) ? 1: 0 X: neighbouring macroblock	0
CtxCbpChroma	1	CtxVarSpat2	4 + (CodedBlockPatternChroma(X) == 2) ? 1: 0 X: neighbouring macroblock	0
CtxCodedBlock	0	CtxVarSpat2	coded_block_flag(X) X: neighbouring block of same BlockType	1, if X is INTRA; 0, otherwise
CtxMbQpDelta	0	CtxVarSpat3	(mb_qp_delta(X) != 0) ? 1: 0 X: neighbouring macroblock	0

Table 9-39 contains the specification of CondTerm for context identifiers and BinIdx relying on templates using two neighbouring symbols. Note that the conditioning term CondTerm of the context identifier CtxCodedBlock depends on six block types as given in Table 9-41 (Luma-DC, Luma-AC, Chroma-U-DC, Chroma-V-DC, Chroma-U-AC, Chroma-V-AC). The interpretation of a neighbouring macroblock used for determination of CtxMbSkip has to be specified more precisely for the case of MB-adaptive frame/field coding. Since the decoder may not know the specific choice of frame or field mode when decoding the mb_skip_flag, it is always assumed that co -location of macroblocks for context determination has to be interpreted as co -location in the sense of pure frame coding. That means, for instance, that for context determination the macroblock on top of the current skipped macroblock in a (potential) top field position is always chosen as the bottom field MB of the MB pair above the current macroblock.

The specification of CtxIdxInc for the context identifiers CtxAbsMvdHoriz and CtxAbsMvdVert for BinIdx equal to 0 is given as follows.

$$CtxIdxInc(CtxAbsMvdComp) = \begin{cases} 0, (AbsMvdComp(A) + AbsMvdComp(B)) < 3, \\ 2, (AbsMvdComp(A) + AbsMvdComp(B)) > 32, \\ 1, otherwise, \end{cases} \quad (9\text{-}5)$$

where Comp denotes the horizontal component (Comp=Horiz) or vertical component (Comp=Vert) and A, B denote the neighbouring blocks of the current block to decode as shown in Figure 9-3. AbsMvdComp denote the absolute value of the corresponding component Comp of the motion vector difference. Since the neighbouring blocks A and B may belong to different macroblock partitions, the following principle for identifying the proper neighbouring blocks in Equation 9 -5 shall be applied. First, the motion vector difference is assumed to be given in oversampled form such that each 4x4 block

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

has its own motion vector difference (Mvd). That means, on the one hand, that in case of a neighbouring block having a coarser partition, the related 4x4 sub-blocks are assumed to inherit the Mvd from the corresponding parent block(s) in the quadtree partition. On the other hand, if the current block C represents a larger block than a 4x4 block, C shall be represented by the corresponding leftmost 4x4 sub-block in the top row of 4x4 sub-blocks. Then, given a block C, the neighbouring 4x4 sub-blocks B on top and A to the left of the representing 4x4 sub-block of C are chosen for evaluation of Equation 9-5. [Ed.Note: This paragraph should be re-written using the geometrical concepts]

9.3.3.1.2 Assignment process of CtxIdxInc for context templates using prior decoded bin values

Let $(b_0,b_1,b_2,\ldots, b_{k-1})$ denote the current bitstring of decoded bins related to a given syntax element C. Then, a generic type of context template associated to BinIdx = k of C is specified as

$$CtxVarBin = CondTerm(b_0,\ldots,b_{k-1}) \qquad (9\text{-}6)$$

where $k \geq 1$. Table 9-40 contains the specification of CondTerm for context identifiers and BinIdx using this type of context template.

Table 9-40 – Specification of CondTerm for context identifiers and BinIdx using the context template according to Equation 9-6

Context identifier	BinIdx	CondTerm($b_0,\ldots,b_{k-1}$)
CtxMbTypeI	4	(b_3 != 0) ? 5: 6
CtxMbTypeI	5	(b_3 != 0) ? 6: 7
CtxMbTypeI	6	7
CtxMbTypeSISuffix	4	(b_3 != 0) ? 5: 6
CtxMbTypeSISuffix	5	(b_3 != 0) ? 6: 7
CtxMbTypeSISuffix	6	7
CtxMbTypeP	2	(b_1 != 0) ? 2: 3
CtxMbTypePSuffix	3	(b_2 != 0) ? 2: 3
CtxMbTypeB	2	(b_1 != 1) ? 4: 5

9.3.3.1.3 Assignment process of CtxIdxInc for context identifiers used for transform coefficients

Three different additional context identifiers are used for conditioning of information related to transform coefficients. All these three types depend on context categories of different block types (BlockType) denoted by the variable CtxCat. The specification of these context categories is given in Table 9-41.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-41 – Context categories for the different block types

BlockType	MaxNumCoeff	CtxCat
Luma DC block for INTRA16x16 mode	16	0:Luma-Intra16-DC
Luma AC block for INTRA16x16 mode	15	1:Luma-Intra16-AC
Luma block for INTRA 4x4 mode	16	2:Luma-4x4
Luma block for INTER 4x4 mode	16	
U-Chroma DC block for INTRA mode	4	3:Chroma-DC
V-Chroma DC block for INTRA mode	4	
U-Chroma DC block for INTER mode	4	
V-Chroma DC block for INTER mode	4	
U-Chroma AC block for INTRA mode	15	4:Chroma-AC
V-Chroma AC block for INTRA mode	15	
U-Chroma AC block for INTER mode	15	
V-Chroma AC block for INTER mode	15	

The context identifiers CtxSigCoeff and CtxLastCoeff are related to significant_coeff_flag and last_significant_coeff_flag, respectively; the CtxIdxInc of the related context identifiers depend on the scanning position ScanningPos within the regarded block:

$$\text{CtxIdxInc}(\text{CtxSigCoeff}, \text{ScanningPos}) = \text{ScanningPos} \tag{9-7}$$

$$\text{CtxIdxInc}(\text{CtxLastCoeff}, \text{ScanningPos}) = \text{ScanningPos} \tag{9-8}$$

For decoding of coeff_absolute_value_minus1, the CtxIdxInc for the context i dentifier CtxAbsLevel is specified depending on BinIdx as follows:

$$\text{CtxIdxInc}(\text{CtxAbsLevel}, \text{BinIdx}=0) = ((\text{NumDecodAbsLevelGt1} != 0)\ ?\ 4 :\ \min(3, \text{NumDecodAbsLevelEq1})) \tag{9-9}$$

$$\text{CtxIdxInc}(\text{CtxAbsLevel}, \text{BinIdx} > 0) = \min(4, \text{NumDecodAbsLevelGt1}) \tag{9-10}$$

where NumDecodAbsLevelEq1 denotes the accumulated number of decoded coefficients with absolute value equal to 1, and NumDecodAbsLevelGt1 denotes the accumulated number of d ecoded coefficients with absolute value greater than 1. Both numbers are related to the same transform coefficient block, where the current decoding takes place, which means that no additional information outside the regarded transform coefficient block is used for the context identifier CtxAbsLevel.

9.3.3.2 Arithmetic decoding process

Input to this process is CtxEqProbFlag and CtxIdx as derived in subclause 9.3.3.1.

Output of this process is the value of the bin.

NOTE - Arithmetic coding is based on the principle of recursive interval subdivision. Given a probability estimation p(0) and p(1)=1−p(0) of a binary decision (0, 1), an initially given code sub-interval with the range CodeIntRange will be subdivided into two sub-intervals having range p(0) × CodeIntRange and CodeIntRange −p(0) × CodeIntRange, respectively. Depending on the decision, which has been observed, the corresponding sub-interval will be chosen as the new code interval, and a binary code string pointing into that interval will represent the sequence of observed binary decisions. It is useful to distinguish between the most probable symbol (MPS) and the least probable symbol (LPS), so that binary decisions have to be identified as either MPS or LPS, rather than 0 or 1. Given this terminology, each context is specified by the probability p_{LPS} of the LPS and the value of MPS, which is either 0 or 1.

The arithmetic core engine in this Recommendation | International Standard has three distinct properties:

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

- The probability estimation is performed by means of a finite-state machine with a table-based transition process between 64 different representative probability states $\{P_k \mid 0 \le k < 64\}$ for the LPS probability p_{LPS}.
- The range CodeIntRange representing the state of the coding engine is quantized to a small set $\{Q_1,\ldots,Q_4\}$ of pre-set quantisation values prior to the calculation of the new interval range. Storing a table containing all 64×4 pre-computed product values of $Q_i \times P_k$ allows a multiplication-free approximation of the product CodeIntRange × P_k.
- For syntax elements or parts thereof with an approximately uniform probability distribution a separate simplified encoding and decoding path is used.

9.3.3.2.1 Probability estimation process

Input to this process are the current State and MPS values of CtxIdx.

Output of this process are the updated State and MPS of the context associated to CtxIdx.

The probability estimator is realized by a set of representative probabilities $\{ P_k \mid 0 \le k < 64 \}$ for the LPS together with some appropriately specified state transition rules. Table 9-42 shows the transition rules for adapting to a given MPS or LPS decision. For transition from one state to another each state is only addressed by its index State, which will be appropriately changed to a new index NextStateMPS(State) or NextStateLPS(State) after the decoding of a MPS or LPS symbol, respectively.

The numbering of the states is arranged in such a way that the state with index State=0 corresponds to a LPS probability value of 0.5, with decreasing LPS probability towards higher states.

After encoding or decoding a decision, an update of the probability estimate is obtained by switching the state index State to a new index:

```
if( decision  = =  MPS )
      State ← NextStateMPS(State)
else
      State ← NextStateLPS(State).
```

In the case, where the current state corresponds to a probability value of 0.5, which corresponds to the State index of 0, and a LPS symbol is observed, the sense of MPS and LPS has to be interchanged.

Table 9-42 – Probability transition

State	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NextStateLPS	0	0	1	2	2	4	4	5	6	7	8	9	9	11	11	12
NextStateMPS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
State	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
NextStateLPS	13	13	15	15	16	16	18	18	19	19	21	21	22	22	23	24
NextStateMPS	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
State	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
NextStateLPS	24	25	26	26	27	27	28	29	29	30	30	30	31	32	32	33
NextStateMPS	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
State	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
NextStateLPS	33	33	34	34	35	35	35	36	36	36	37	37	37	38	38	63
NextStateMPS	49	50	51	52	53	54	55	56	57	58	59	60	61	62	62	63

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table 9-43 – RTAB[State][QuantCodeIntRange] table for interval subdivision

State	Q				State	Q			
	0	1	2	3		0	1	2	3
0	128	176	208	240	32	27	33	39	45
1	128	167	197	227	33	26	31	37	43
2	128	158	187	216	34	24	30	35	41
3	123	150	178	205	35	23	28	33	39
4	116	142	169	195	36	22	27	32	37
5	111	135	160	185	37	21	26	30	35
6	105	128	152	175	38	20	24	29	33
7	100	122	144	166	39	19	23	27	31
8	95	116	137	158	40	18	22	26	30
9	90	110	130	150	41	17	21	25	28
10	85	104	123	142	42	16	20	23	27
11	81	99	117	135	43	15	19	22	25
12	77	94	111	128	44	14	18	21	24
13	73	89	105	122	45	14	17	20	23
14	69	85	100	116	46	13	16	19	22
15	66	80	95	110	47	12	15	18	21
16	62	76	90	104	48	12	14	17	20
17	59	72	86	99	49	11	14	16	19
18	56	69	81	94	50	11	13	15	18
19	53	65	77	89	51	10	12	15	17
20	51	62	73	85	52	10	12	14	16
21	48	59	69	80	53	9	11	13	15
22	46	56	66	76	54	9	11	12	14
23	43	53	63	72	55	8	10	12	14
24	41	50	59	69	56	8	9	11	13
25	39	48	56	65	57	7	9	11	12
26	37	45	54	62	58	7	9	10	12
27	35	43	51	59	59	7	8	10	11
28	33	41	48	56	60	6	8	9	11
29	32	39	46	53	61	6	7	9	10
30	30	37	43	50	62	6	7	8	9
31	29	35	41	48	63	2	2	2	2

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

9.3.3.2.2 Table-based arithmetic decoding process

Input to this process are CtxIdx and BinIdx, for which the arithmetic decoding proce ss is invoked, CtxEqProbFlag, and the state variables CodeIntRange and CodeIntValOffset of the arithmetic decoding engine.

Output of this process is the value of the bin corresponding to BinIdx.

The status of the arithmetic decoding engine is represented b y the range of the code sub-interval (CodeIntRange) and the variable CodeIntValOffset specifiying the offset between a value pointing into the code sub -interval and the lower bound of the corresponding code sub-interval. Figure 9-4 illustrates the whole decoding process. By invoking the InitDecoder process before starting the parsing of the slice data, as specified in subclause 9.3.1.2, CodeIntRange and CodeIntValOffset are initialised. For decoding of th e value of a bin for a given BinIdx and CtxIdx, the context index CtxIdx is passed to the actual arithmetic decoding processes Decode(CtxIdx), Decode_eq_prob(CtxIdx) or Decode_terminate(CtxIdx). If CtxEqProbFlag is equal to 1, Decode_eq_prob( ) is invoked. If CtxEqProbFlag is equal to 0 and CtxIdx is equal to 253, Decode_eq_prob( ) is invoked; in all other cases Decode( ) as specified in subclause 9.3.3.2.2.1 shall be applied.

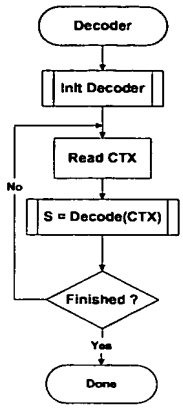


Figure 9-4 – Overview of the arithmetic decoding process (informative) [Ed.Note: Either Flowchart has to be updated or removed]

9.3.3.2.2.1 Decoding process for a decision

Input to this process are CtxIdx and the state variables CodeIntRange and CodeInt ValOffset.

Output of this process are the decoded value, and the updated state variables CodeIntRange and CodeIntValOffset.

Figure 9-5 shows the flowchart for decoding a single decision. In a first step, the estimation of the sub -interval ranges CodeIntRange(LPS) and CodeIntRange(MPS) = CodeIntRange – CodeIntRange(LPS) corresponding to the LPS and the MPS decision is performed as follows.

Given the interval range CodeIntRange, the CodeIntRange is first mapped to a quantized value Qu antCodeIntRange according to

$$\text{QuantCodeIntRange} = (\text{CodeIntRange} >> 6) - 4 \qquad (9\text{-}11)$$

such that the state index State and QuantCodeIntRange are used as an entry in the look-up table RTAB to determine CodeIntRange(LPS):

$$\text{CodeIntRange(LPS)} = \text{RTAB}[\text{State}][\text{QuantCodeIntRange}]. \qquad (9\text{-}12)$$

Table 9-43 specifies the corresponding values of RTAB.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

In a second step, the current value of CodeIntValOffset is compared to the size of the MPS sub-interval CodeIntRange ← CodeIntRange – CodeIntRange(LPS). If CodeIntValOffset is greater than or equal to CodeIntRange a LPS is decoded, CodeIntValOffset is decremented by CodeIntRange and the new range CodeIntRange is s et to CodeIntRange(LPS); otherwise a MPS is decoded. Given the decoded decision, the probability update is performed accordingly as specified in subclause 9.3.3.2.1. Depending on the current value of the new range CodeIntRange, renormalization will be performed as specified in subclause 9.3.3.2.2.2.

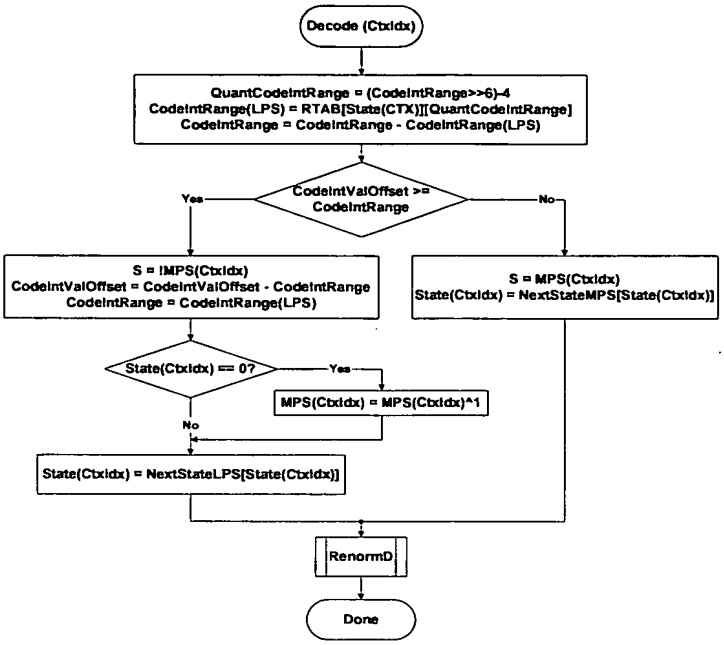


Figure 9-5 – Flowchart for decoding a decision

9.3.3.2.2.2 Renormalization process in the arithmetic decoding engine

Input to this process are bits from a slice layer RBSP and the state variables CodeIntRange and CodeIntValOffset.

Output of this process are the updated state variables CodeIntRange and CodeIntValOffset.

Renormalization is illustrated in Figure 9-6. The current range CodeIntRange is first logically compared to 0x0100: If it is greater than or equal to that value, no renormalization is needed and RenormD is finished; otherwise the renormalization loop is entered. Within this loop, the range CodeIntRange is doubled, i.e. left-shifted by 1 and a single bit is shifted into CodeIntValOffset by using read_bits(1).

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

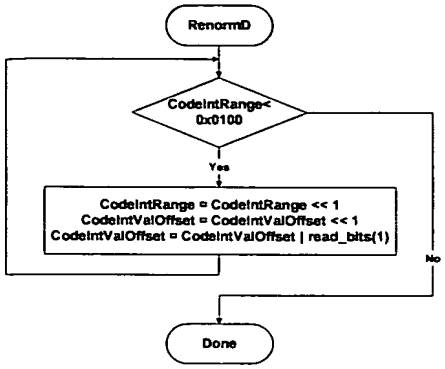


Figure 9-6 – Flowchart of renormalization

9.3.3.2.2.3 Arithmetic decoding process for decisions with uniform pdf using arithmetic decoder bypass

Input to this process are bits from a slice layer RBSP and the state variables CodeIntRange and CodeIntValOffset.

Output of this process are the updated state var iables CodeIntRange and CodeIntValOffset, and the decoded value.

This special decoding routine is invoked when CtxEqProbFlag is equal to 1. Figure 9-7 specifies the decoding process for this special case.

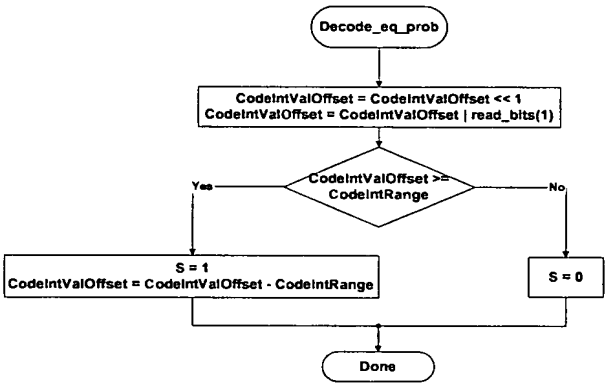


Figure 9-7 – Flowchart of decoding bypass

9.3.3.2.2.4 Decoding process for decisions before termination

Input to this process are bits from a slice layer RBSP and the state variables CodeIntRange and CodeIntValOffset.

Output of this process are the updated state variables CodeIntRange and CodeIntValOffset, and the decoded value.

This special decoding routine applies to decoding of end_of_slice_flag and of the bin indicating the PCM mode. Probabilities are fixed and not adapted. Figure 9-8 shows the flowchart of the corresponding decoding process. When the decision S is equal to 1 no renormalization is carried out and CABAC decoding is terminated. In such a case the last bit inserted in register CodeIntValOffset is the stop_bit.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

NOTE – This procedure may also be implemented using decode(CtxIdx) and setting State(CtxIdx) to 63 and MPS(CtxIdx) to 0. In the case where the result is S = 1, seven more bits would be read by decode(CtxIdx) and a decoder would have to adjust its bitstream pointer accordingly to properly decode following syntax elements.

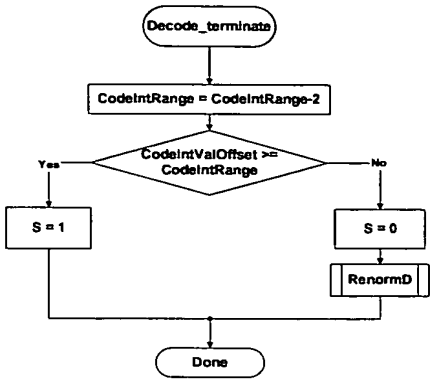


Figure 9-8 – Flowchart of decoding a decision before termination

9.3.3.2.3 Table-based arithmetic encoding process (informative)

[Ed.Note: This subclause needs to be revisited and updated according to the new terminology used in the arithmetic decoding spec.]

This subclause does not form an integral part of this Recommendation | International Standard.

Input to this process are decisions that are to be encoded and written, and the arithmetic encoding engine variables R and L.

Output of this process are bits that are written to the RBSP.

This informative subclause describes an arithmetic encoding engine that mach es the arithmetic decoding engine described in subclause 9.3.3.2.2. The status of the arithmetic encoding engine is represented by a value L pointing to the lower end of a sub-interval and the corresponding range R of that sub -interval. Figure 9-9 gives an illustration of the whole encoding process. Performing the InitEncoder procedure, which is further specified in subclause **Error! Reference source not found.**, L and R will be appropriately initialised. For encoding of each single decision S, the following two-step operation is employed: first, the related context model CTX is determined according to the rules specified in subclauses **Error! Reference source not found.**. Given the context model CTX and a symbol S the encoding operation Encode(CTX,S) then writes zero or more bits as described in subclause 9.3.3.2.3.1 to the current RBSP.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

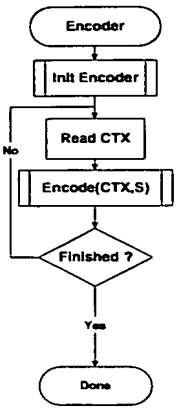


Figure 9-9 – Overview of the arithmetic encoding process

9.3.3.2.3.1 Encoding process for a decision (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Input to this process are the context variable (CTX), the decision to be encoded (S), and the arithmetic coding engine variables R and L.

Output of this process is the state of the arithmetic encoding engine State(CTX).

Figure 9-10 shows the flowchart for encoding a single decision. In a first step, the estimation of the sub-interval ranges R_{LPS} and $R_{MPS} = R - R_{LPS}$ corresponding to the LPS and the MPS decision is performed as follows.

Given the interval range R, we first map R to a quantized value Q using Equation 9-11 such that the state index State and Q are used as an entry in the look-up table RTAB to determine R_{LPS} as specified in Equation 9-12.

In a second step, the value of *S* is compared to the value of MPS(CTX). If S is different from MPS(CTX), R is added to L and then R is set to the value R_{LPS}. Given the encoded decision, the probability update is performed accordingly as specified in subclause 9.3.3.2.1. Depending on the current value of the new range R, renormalization will be performed as described in more detail in subclause 9.3.3.2.3.2.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

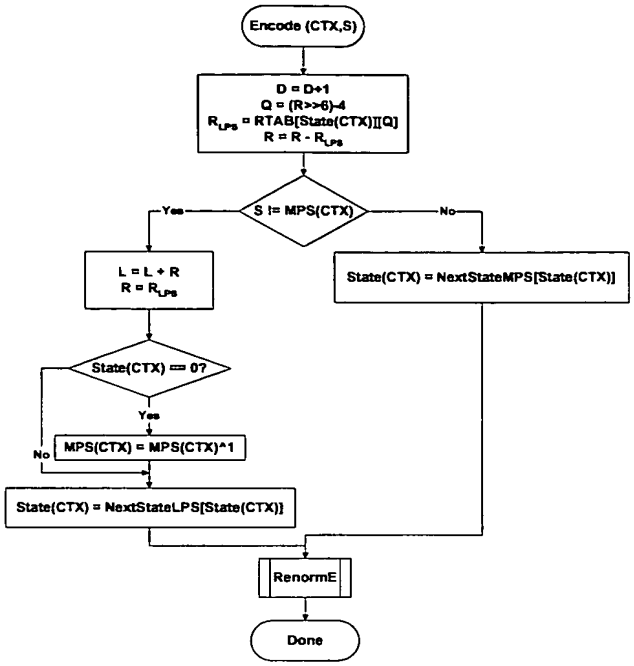


Figure 9-10 – Flowchart for encoding a decision

9.3.3.2.3.2 Renormalization process in the arithmetic encoding engine (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Input to this process are arithmetic coding engine variables R and L.

Output of this process are bits that are written to the RBSP.

Renormalization is illustrated in Figure 9-11. The current range *R* is first logically compared to 0x0100: If it is greater than or equal to that value, no renormalization is needed and RenormE is finished; otherwise the renormalizat ion loop is entered. Within this loop, it is first determined if a 0 or a 1 can safely be output, e.g. there is no possibility for a carry over in a future iteration. Otherwise the BO register is incremented by 1. Finally the values of L and R are doubled. The PutBit procedure described in Figure 9-12 provides carry over control.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

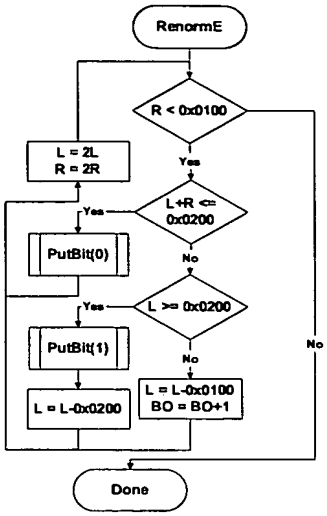


Figure 9-11 – Flowchart of renormalization

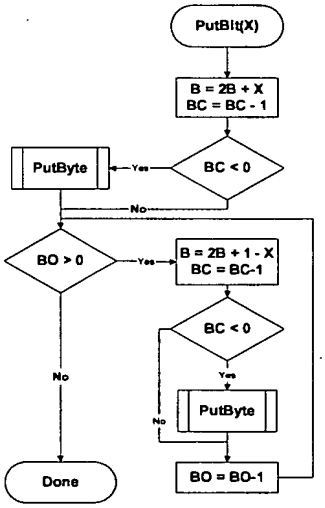


Figure 9-12 – Flowchart of putting bit

9.3.3.2.3.3 Writing process for compressed bytes (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Input to this process is/are ??? [Ed.Note: spec ify input to this process]

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Output of this process is/are ??? [Ed.Note: specify output of this process]

Figure 9-13 shows how the output of compressed data is performed. First, a new byte of compressed data is written out to the RBSP bitstream C; then the index CL pointing to the current position of the bitstream C is incremented by 1 and the bit-counter is set to 7.

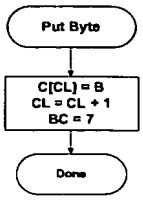


Figure 9-13 – Flowchart for output of compressed bytes

9.3.3.2.3.4 Arithmetic encoding process for decisions with uniform pdf using encoder bypass (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Input to this process is/are ??? [Ed.Note: specify input to th is process]

Output of this process is/are ??? [Ed.Note: specify output of this process]

This special encoding routine applies to encoding of the sign information of motion vector data and the sign of the levels of significant transform coefficients, which are assumed to have a uniform probability distribution. Renormalization is included in the Figure 9-14.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

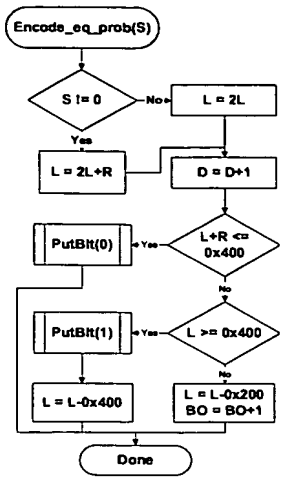


Figure 9-14 – Flowchart of encoding bypass

9.3.3.2.3.5 Encoding process for a decision before termination (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Input to this process is/are ??? [Ed.Note: specify input to this process]

Output of this process is/are ??? [Ed. Note: specify output of this process]

This encoding routine shown in Figure 9-15 applies to encoding of end_of_slice_flag and of the bin indicating the PCM mode. Probabilities are fixed and not adapted. When the symbol S is equal t o 1 CABAC encoding is terminated. In such a case the last byte B written by PutByte( ) contains the stop_bit as well as the alignment bits.

After terminating the arithmetic encoder at the end of a slice, the encapsulation procedure is invoked. Let N be th e number of bytes in the NAL unit after encapsulation and M the number of macroblocks in the slice. Furthermore let K be ceil((ceil((D-M)/32)-N)/3). If K is smaller than or equal to zero, no cabac_stuffing_bytes are appended to the NAL unit. Otherwise the 3-byte sequence 0x000003 is appended K times to the NAL unit after encapsulation, where the first two bytes 0x0000 represent a cabac_stuffing_word and the third byte 0x03 represents an emulation_prevention_byte.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

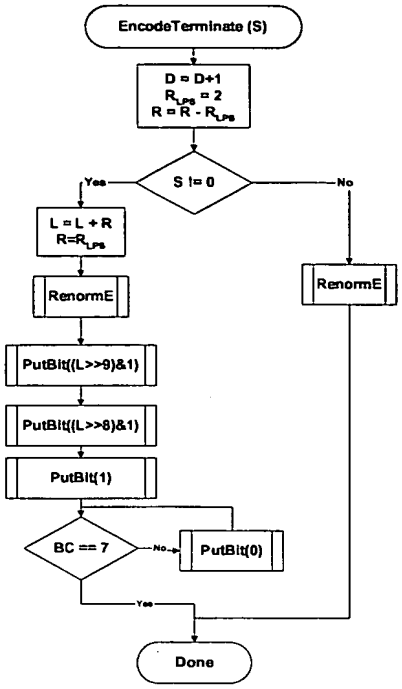


Figure 9-15 – Flowchart of encoding a decision before termination

Annex A
Profiles and levels

(This annex forms an integral part of this Recommendation | International Standard)

[Ed. Note:

1. Depending on final terminology, it may be necessary to change "picture" to "frame"; check terms. "Max picture size" is meant to refer to a complete picture (both fields) if interlaced. The same goes for other uses of the word "picture" in this annex.
2. Yellow highlighted text in this subclause re quires further consideration regarding proper terminology.
3. We need limits on slice sizes to properly bound slice_id]

Profiles and Levels specify the capabilities needed to decode the bitstream, and may be used to indicate interoperability points between individual decoder implementations.

NOTE - This Recommendation | International Standard does not include individually selectable "options" at the decoder, as this would increase interoperability difficulties.

Each Profile specifies a set of algorithmic featu res and limits which shall be supported by all decoders conforming to that Profile. Note that encoders are not required to make use of any particular set of features supported in a Profile.

Each Level specifies a set of limits on the values which may be t aken by the parameters of this Recommendation | International Standard. The same set of Level definitions is used with all Profiles, but individual implementations may support a different Level for each supported Profile. For any given Profile, Levels ge nerally correspond to decoder processing and memory capability, in units based on video decoding, rather than on specific implementation platforms.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

A.1 Requirements on video decoder capability

Capabilities of video decoders conforming to this Recommendation | International Standard are specified in terms of the ability to decode video streams conforming to constraints Profiles and Levels specified in this Annex. For each such Profile, the Level supported for that Profile shall also be expressed. Such expressi on may be in the form of coded values equivalent to a specific Profile and specific Level from this Annex.

Specific values are specified in this annex for the syntax elements profile_idc and level_idc. All other values of profile_idc and level_idc are reserved for future use by ITU-T | ISO/IEC.

NOTE: Decoders should not infer that if a reserved value of profile_idc or level_idc falls between the values specified in this Recommendation | International Standard this indicates intermediate capabilities between the specified profiles or levels, as there are no restrictions on the method to be chosen by ITU-T | ISO/IEC for the use of such future reserved values.

A.2 Profiles

A.2.1 Baseline profile

[Ed.Note: this subclause needs to be formalized s.t. the use of a feature i s indicated by a values of syntax elements.]

All decoders supporting the Baseline Profile shall be capable of decoding bitstreams which use the following features:

a) I and P slices types
b) Deblocking filter
c) Pictures with field_pic_flag equal to 0
d) Pictures with mb_adaptive_frame_field_flag equal to 0
e) Zig-zag scan
f) 1/4-sample inter prediction
g) Tree-structured motion segmentation down to 4x4 block size
h) VLC-based entropy coding
i) Arbitrary slice order (ASO): In Baseline profile, the decoding order of slices within a pi cture may not follow the constraint that first_mb_in_slice is monotonically increasing within the NAL unit stream for a picture.
j) Flexible macroblock ordering (FMO, maximum 8 slice groups): In Baseline profile, num_slice_groups_minus1<8.
k) Redundant slices
l) 4:2:0 chroma format

Conformance to the Baseline Profile is indicated by setting the syntax element profile_idc equal to 66.

A.2.2 Main profile

All decoders supporting the Main Profile shall be capable of decoding bitstreams which use the following features:

a) B slice types
b) CABAC
c) Weighted prediction
d) All features included in the Baseline Profile except:
    1. Arbitrary Slice Order (ASO): In Main profile, the decoding order of slices within a picture shall follow the constraint that first_mb_in_slice shall be monotonically i ncreasing within the NAL unit stream for a picture.
    2. Flexible Macroblock Order (FMO): In Main profile, num_slice_groups_minus1 shall be zero.
    3. Redundant Slices

Decoders supporting the Main Profile Level 2.1 and above shall also be capable of decoding bitstr eams using:

e) Pictures with field_pic_flag equal to 1
f) Pictures with mb_adaptive_frame_field_flag equal to 1

Conformance to the Main profile is indicated by setting the syntax element profile_idc equal to 77.

A.2.3 X profile

All decoders supporting the X Profile s hall be capable of decoding bitstreams which use the following features:

a) B slice types
b) SP and SI slice types
c) Slice slice data partitioning (nal_unit_type being equal to 2 through 4)
d) Weighted prediction
e) All features included in the Baseline Profile

Decoders supporting the X Profile Level 2.1 and above shall also be capable of decoding bitstreams using:

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

f) Pictures with field_pic_flag equal to 1
g) Pictures with mb_adaptive_frame_field_flag equal to 0 [Ed.Note: there was a 1 before in versions d16 to d36, my mistake, I believe]

All video decoders supporting the X Profile shall also support the Baseline Profile. The Level number supported for the Baseline Profile shall not be less than the Level number supported for the X Profile.

Conformance to the X Profile is indicated by setting the syntax element profile_idc equal to 88.

A.3 Levels

Level limits are expressed in units of whole luma macroblocks. If a particular picture height or width is not an exact multiple of a whole macroblock, that dimension shall be considered as rounded up to the next whole macroblock for the purposes of conformance with this subclause.

The definition of support for a given Level is that any picture size/frame rate combination shall be decoded where the:

a) sample processing rate (in whole macroblocks/second) is <= the Level limit given, and,

b) (frame_width_in_mbs_minus1+1) * (frame_height_in_mbs_minus1+1) <= MaxFS , and,

c) frame_width_in_mbs_minus1+1 <= sqrt(MaxFS * 8), and,

d) frame_height_in_mbs_minus1+1 <= sqrt(MaxFS * 8), and,

e) The the total number of bytes of NAL unit data (i.e., the total of the NumBytesInNALunit variables for the picture) that contain primary slices and slice data partitions for any coded frame is <= 128 * MaxFS * ChromaFormatParameter.

f) The the total number of bytes of NAL unit data (i.e., the total of the NumBytesInNALunit variables for the picture) that contain primary slices and slice data partitions for any coded field is <= 64 * MaxFS * ChromaFormatParameter

g) the HRD DPB verifier (subclause C.1.2) is satisfied with DPBcapacity as defined in subclause A.6.3, and,

h) maximum video bit rate of the bitstream is <= the Level limit given, and,

i) required CPB buffer size is <= the Level limit given, and,

j) vertical motion vector range does not exceed the Level limits given, and,

k) horizontal motion vector range does not exceed [-2048, 2047.75] in full-sample luma displacement units, and

l) luma direct and bi-predictive block size does not exceed the Level limits given, and

m) number of motion vectors per two consecutive macroblocks in decoding order (including from the last macroblock of a slice to the first macroblock of the next slice) does not exceed the Level limits given, and

n) picture rate is not greater than 172 frames per second (with each field picture counting as one half of a frame for purposes of this computation), and

o) number of reference frames is not greater than 15, and

p) number of sequence parameter sets is not greater than 32, and

q) number of picture parameter sets is not greater than 128, and

r) number of bits of macroblock_layer( ) data for any macroblock is not greater than 3200.

The definition of each Level includes the requirements of all lower numbered Levels in Table A-1. Decoders supporting a given Level shall also be capable of decoding bitstreams using all lower numbered Levels.

Note - display of decoded video is outside the scope of this Recommendation | International Standard; some decoder implementations may not include displays at all, and display limitations do not necessarily cause interoperability failures. [Ed. Note: why is this note here? This sort of thing is already covered elsewhere.]

"Picture size" means the total number of macroblocks in the complete picture (both even and odd fields if interlaced).

A.3.1 Level limits

Table A-1 below gives the parameter limits for each Level. Entries marked " -" in Table A-1 denote the absence of a corresponding limit.

Conformance to a particular Level shall be indicated by setting the syntax element level_idc equal to a value of ten times the level number specified in Table A-1.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table A-1 – Level Limits

Level #	Max Sample Processing Rate (MB/s)	Max Frame Size MaxFS (MBs)	Decoded picture buffer memory MaxKB (1024 bytes)	Max Video Bitrate (1000 bits/sec)	Max CPB Size (1000 bits)	Vertical MV Range (full samples)	Minimum luma Direct & Bi-predictive block size	Maximum number of MVs per two consecutive MBs
1	1 485	99	148.5	64	175	[-64,+63.75]	-	-
1.1	2 970	396	891.0	128	325	[-128,+127.75]	-	-
1.2	5 940	396	891.0	768	2 000	[-128,+127.75]	-	-
2	11 880	396	891.0	2 000	2 000	[-128,+127.75]	-	-
2.1	19 800	792	1 782.0	4 000	4 000	[-256,+255.75]	-	-
2.2	20 250	1 620	3 037.5	4 000	4 000	[-256,+255.75]	-	-
3	40 500	1 620	3 037.5	8 000	8 000	[-256,+255.75]	-	-
3.1	108 000	3 600	6 750.0	20 000	14 000	[-512,+511.75]	8x8	16
3.2	216 000	5 120	7 680.0	20 000	20 000	[-512,+511.75]	8x8	16
4	245 760	8 192	12 288.0	20 000	25 000	[-512,+511.75]	8x8	16
5	786 432	32 768	61 440.0	150 000	450 000	[-512,+511.75]	8x8	16

Levels with non-integer Level numbers in Table A-1 are refered to as "intermediate Levels". All Levels have the same status, but note that some applications may choose to use only the integer -numbered Levels.

The maximum video bitrate and maximum coded picture buffer (CPB) size in Table A-1 apply to VCL conformance. The equivalent limits for NAL conformance are 20% higher than these values. [Ed. Note: Do we have a clear definition of what distinguishes VCL from NAL?]

Informative subclause A.3.3 shows the effect of these limits on frame rates for several example picture formats.

A.3.2 Decoded picture buffer capacity

For verification by the HRD DPB verifier, the capacity of the DPB in frames for a given mode is defined. The memory availible for the DPB is defined in Table A-1. The capacity of the DPB is calculated by:

[Ed. Note (Karsten): Subclause 6.2 defines 4:2:0 as the only chroma format in H.264, so do why we need this ChromaFormatParameter?]

[Ed.Note: maybe we should add a syntax element (ue(v)) to make this future proof]

DPBcapacity = Min(max_dec_frame_buffering, 1024 * MaxKB / ((frame_width_in_mbs_minus1+1) * (frame_height_in_mbs_minus1+1) * (256*ChromaFormatParameter))).

The parameter DPBcapacity is in units of frames.

The parameter ChromaFormatParameter shall take values according to Table A-2:

Table A-2 – ChromaFormatParameter values

Chroma Format	ChromaFormatParameter
monochrome	1
4:2:0	1.5
4:2:2	2
4:4:4	3

A.3.3 Effect of level limits on frame rate (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table A-3 – Maximum frame rates for some common picture sizes (Hz)

Level number:					1	1.1	1.2	2	2.1	2.2	3	3.1	3.2	4	5
Max picture size (macroblocks):					99	396	396	396	792	1,620	1,620	3,600	5,120	8,192	32,768
Max macroblocks/second:					1 485	2 970	5 940	11 880	19 800	20 250	40 500	108 000	216 000	245 760	786 432
Max picture size (samples):					25 344	101 376	101 376	101 376	202 752	414 720	414 720	921 600	1 310 720	2 097 152	8 388 608
Max samples/second:					380 160	760 320	1 520 640	3 041 280	5 068 800	5 184 000	10 368 000	27 648 000	55 296 000	62 914 560	201 326 592
	Samples			Luma											
Format	Width	Height	MBs	Samples											
SQCIF	128	96	48	12 288	30.9	61.9	123.8	172.0	172.0	172.0	172.0	172.0	172.0	172.0	172.0
QCIF	176	144	99	25 344	15.0	30.0	60.0	120.0	172.0	172.0	172.0	172.0	172.0	172.0	172.0
QVGA	320	240	300	76 800	-	9.9	19.8	39.6	66.0	67.5	135.0	172.0	172.0	172.0	172.0
525 SIF	352	240	330	84 480	-	9.0	18.0	36.0	60.0	61.4	122.7	172.0	172.0	172.0	172.0
CIF	352	288	396	101 376	-	7.5	15.0	30.0	50.0	51.1	102.3	172.0	172.0	172.0	172.0
525 HHR	352	480	660	168 960	-	-	-	-	30.0	30.7	61.4	163.6	172.0	172.0	172.0
625 HHR	352	576	792	202 752	-	-	-	-	25.0	25.6	51.1	136.4	172.0	172.0	172.0
VGA	640	480	1 200	307 200	-	-	-	-	-	16.9	33.8	90.0	172.0	172.0	172.0
525 4SIF	704	480	1 320	337 920	-	-	-	-	-	15.3	30.7	81.8	163.6	172.0	172.0
525 SD	720	480	1 350	345 600	-	-	-	-	-	15.0	30.0	80.0	160.0	172.0	172.0
4CIF	704	576	1 584	405 504	-	-	-	-	-	12.8	25.6	68.2	136.4	155.2	172.0
625 SD	720	576	1 620	414 720	-	-	-	-	-	12.5	25.0	66.7	133.3	151.7	172.0
SVGA	800	600	1 900	486 400	-	-	-	-	-	-	-	56.8	113.7	129.3	172.0
XGA	1024	768	3 072	786 432	-	-	-	-	-	-	-	35.2	70.3	80.0	172.0
720p	1280	720	3 600	921 600	-	-	-	-	-	-	-	30.0	60.0	68.3	172.0
4VGA	1280	960	4 800	1 228 800	-	-	-	-	-	-	-	-	45.0	51.2	163.8
SXGA	1280	1024	5 120	1 310 720	-	-	-	-	-	-	-	-	42.2	48.0	153.6
525 16SIF	1408	960	5 280	1 351 680	-	-	-	-	-	-	-	-	-	46.5	148.9
16CIF	1408	1152	6 336	1 622 016	-	-	-	-	-	-	-	-	-	38.8	124.1
4SVGA	1600	1200	7 500	1 920 000	-	-	-	-	-	-	-	-	-	32.8	104.9
1080i	1920	1080	8 160	2 088 960	-	-	-	-	-	-	-	-	-	30.1	96.4
2Kx1K	2048	1024	8 192	2 097 152	-	-	-	-	-	-	-	-	-	30.0	96.0
4XGA	2048	1536	12 288	3 145 728	-	-	-	-	-	-	-	-	-	-	64.0
16VGA	2560	1920	19 200	4 915 200	-	-	-	-	-	-	-	-	-	-	41.0

Note 1 This is a variable-picture-size specification. The specific picture sizes in Table A-3 are illustrative examples only.

Note 2 As used in Table A-3, "525" refers to typical use for environments using 525 analogue scan lines (of which approximately 480 lines contain the visible picture region), and "625" refers to environments using 625 analogue scan lines (of which approximately 576 lines cont ain the visible picture region).

Note 3 XGA is also known as (aka) XVGA, 4SVGA aka UXGA, 16XGA aka 4Kx3K, CIF aka 625 SIF, 625 HHR aka 2CIF aka 1/2 625 D-1, aka 1/2 625 ITU-R BT.601, 525 SD aka 525 D-1 aka 525 ITU-R BT.601, 625 SD aka 625 D-1 aka 625 ITU-R BT.601.

Note 4 Frame rates given are correct for progressive scan modes, and for interlaced if "MB High" column value is even.

Annex B
Byte stream format

(This annex forms an integral part of this Recommendation | International Standard)

This annex specifies syntax and semantics of a byte stream format specified for use by systems that transmit some or all of the NAL unit stream as an ordered stream of bytes or bits within which the locations of NAL unit boundaries need to be identifiable from patterns in the data, such as ITU-T Recommendation H.222.0 | ISO/IEC 13818 -1 systems or ITU-T Recommendation H.320 systems. For bit-oriented transmission, the network bit order for the byte stream format is specified to start with the MSB of the first byte, proceed t o the LSB of the first byte, followed by the MSB of the second byte, etc.

The byte stream format consists of a sequence of byte_stream_nal_unit( ) structures. Each byte_stream_nal_unit( ) contains one start code prefix followed by one nal_unit( ). It may (and under some circumstances, it shall) also contain some additional zero_byte syntax elements.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

B.1 Byte stream NAL unit syntax and semantics

B.1.1 Byte stream NAL unit syntax

byte_stream_nal_unit( ) {	C	Mnemonic
while(next_bits(24) != 0x000001)		
zero_byte /* equal to 0x00 */		f(8)
if(more_data_in_byte_stream( )) {		
start_code_prefix /* equal to 0x000001 */		f(24)
nal_unit( )		
}		
}		

B.1.2 Byte stream NAL unit semantics

The order of byte stream NAL units in the byte stream shall follow th e decoding order of the NAL units contained in the byte stream NAL units.

zero_byte is a single byte equal to 0x00.

The minimum required number of zero_byte syntax elements preceding the start_code_prefix is 1 when either of the following conditions are fu lfilled:

- If the nal_unit_type within the nal_unit( ) is equal to 0x07 (sequence parameter set) or 0x08 (picture parameter set), or
- If the byte_stream_nal_unit( ) contains the first nal_unit( ) of a picture in decoding order, specifically defined as the first SEI message, picture delimiter, slice, or slice data partition nal_unit( ) specific to a particular picture in decoding order.

Any number of additional zero_byte syntax elements may immediately precede the start code prefix within the byte_stream_nal_unit( ).

[Ed. Note: Need to clarify scope of each SEI message to identify those that are considered "specific to a particular picture" – Note that this i salso needed for other purposes such as definition of decoding order for SEI messages.]

start_code_prefix is a fixed-value string of 3 bytes equal to 0x000001.

The use of the byte-aligned 3-byte sequence 0x000002 within the byte stream format is reserved for future use by ITU-T | ISO/IEC. [Ed. Note: What is the decoder supposed to do if it sees that valu e in a byte stream? My guess is that it should choke, so maybe we don't need to state that we reserve that byte string.]

B.2 Decoder byte-alignment recovery (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

If the decoder does not have byte alignment with the encoder's byte stream, the decoder can examine the incoming bitstream for the binary pattern '00000000 00000000 00000000 00000001' (31 consecutive bits equal to 0 followed by bit equal to 1). The bit immediately following this pattern is the first bit of an aligned byte. Upon detecting this pattern, the decoder will be byte aligned with the encoder.

Once byte aligned with the encoder, the decoder can examine the incoming byte stream for the three -byte sequences 0x000001 and 0x000003.

If the three-byte sequence 0x000001 is detected, this represents a start code prefix.

If the three-byte sequence 0x000003 is detected, the decoder discards the third byte (0x03) as specified in subclause **Error! Reference source not found.**.

Many systems are inherently byte aligned, and thus have no need for the bit -oriented byte alignment detection procedure described in this subclause.

The byte alignment detection procedure described in this subclause is equivalent to searching a byte sequence for three consecutive zero-valued bytes (0x000000), starting at any alignment position. Detecting this pattern indicates that the next non-zero byte contains the end of a start code prefix, and the fir st non-zero bit in that next non-zero byte is the last bit of an aligned byte.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Annex C
Hypothetical Reference Decoder

(This annex forms an integral part of this Recommendation | International Standard)

This annex specifies ??? [Ed.Note: What ?]

C.1 Hypothetical reference decoder and buffering verifiers

The hypothetical reference decoder (HRD) represents a set of normative requirements on bitstreams. These constraints shall be enforced by an encoder, and can be assumed by a decoder or multiplexor to be true.

This subclause specifies the normative requirements of the HRD. Subclause C.2 provides additional information that is important for a full understanding of the HRD operation.

Two types of bitstreams are subject to the HRD requirements of this Recommendation | International Standard; a stream of VCL NAL units and a NAL bitstream (either a byte stream or a packet stream). Figure C-1 shows how each of these is constructed from the RBSP. In other words, a given set of HRD parameters may pertain to the VCL data only or to the multiplexed combination of VCL and NAL. This is signalled through video usability information (subclauses E.1 and E.2). Although the HRD may apply to the VCL, HRD information is not carried in the VCL, but only in the higher layer, whether a byte stream or a packet stream.

NOTE - When HRD information is contained within the bitstream, it is possible to verify the conformance of a bitstream to the requirements of this subclause based solely on information contained in the bitstream. If the HRD information is not present, as is the case for all stand-alone VCL streams, conformance can only be verified if the HRD data is supplied by external means.

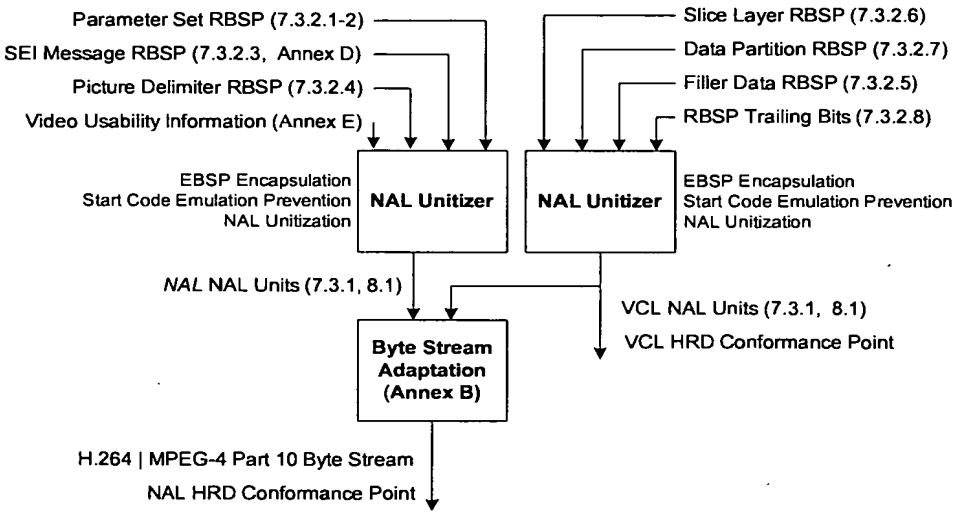


Figure C-1 – Structure of Byte streams and NAL unit streams and HRD Conformance Points

The HRD contains the following buffers, as shown in Figure C-2:

- one or more coded picture buffers (CPB), each of which is either variable bit rate (VBR) or constant bit rate (CBR), and
- one decoded picture buffer (DPB) attached to the output of one of the CPBs.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

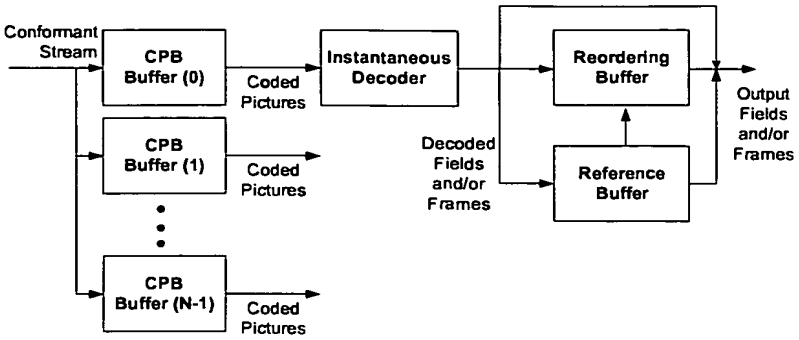


Figure C-2 – HRD Buffer Verifiers

The multiple CPBs exist because a bitstream may conform to multiple CPBs, as detailed in subclause C.2.2.

Considering the operation of a single CPB, data associated with coded pictures flow into the CPB according to a specified arrival schedule. Each coded picture is removed instantaneously and decoded by the decoder at CPB removal times. Decoded pictures are placed in the DPB at the CPB removal time. Finally, pictures are removed from the DPB at the later of the DPB output time and the time that they are no longer needed as reference for decoding. The conditions on a conformant bitstream are given below. The primary conditions are that the CPB neither underflow nor overflow and that the DPB does not overflow.

HRD syntax and semantics is contained in the following subclauses. High -level HRD information concerning the number of CPBs, their bitrates and buffer sizes is found in Annex E under subclauses E.1.1, E.1.2, E.2.1 and E.2.2. This information is part of the video usability information of the sequence parameter set. Timing information needed to initialize the HRD is contained in the buffering period SEI messages (subclauses D.1.1 and D.2.1). Timing information relating to the removal of pictures through the CPB and DPB is found in the picture timing SEI message (subclauses D.1.2 and D.2.2).

All the arithmetic in this annex is done with real values, so that no rounding errors can propagate. For example, the number of bits in a CPB just prior to or after removal of a transmitted picture is not nec essarily an integer. Furthermore, while conformance is guaranteed under the assumption that all frame -rates and clocks used to generate the bitstream match exactly the values signalled in the bitstream, in a real system each of these may vary from the sig nalled or specified value.

In the following description, let t_c = num_units_in_tick ÷ time_scale be defined as the *clock tick*. The clock tick is a time interval no larger than the shortest possible inter -picture capture interval in seconds. Also let $be[t]$ and $te[b]$ be the bit equivalent of a time t and the time equivalent of a number of bits b, with the conversion factor being the CPB arrival bit rate.

C.1.1 Operation of coded picture buffer

[Ed.Note: This description requires clear definition of a random access point.]

This specification applies independently to each set of CPB parameters wit hin the HRD VUI portion of the sequence parameter set. The bits associated with each picture include all NAL units associated with primary and redundant slices of the picture and all the data that follow the last NAL unit of the previous picture, includin g parameter sets, video usability information and SEI messages. This description applies to both NAL and VCL conformance, although for VCL conformance some of these data types will not be present.

C.1.1.1 Timing of bitstream arrival

The HRD may be initialized at the first picture following any random access point, defined by a random access point SEI message. The buffer is initially empty. The first bit of the first picture after the random access SEI message begins to enter the buffer at *initial arrival time* $t_{ai}(0)=0$ at the bit rate bit_rate[k] associated with the CPB (see subclauses E.2.2 and E.3.2). The last bit of the first transmitted picture finishes arriving at *final arrival time*

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

$$t_{af}(0) = b(0) \div \text{bit\_rate}[k], \quad \text{(C-1)}$$

where $b(n)$ is the size in bits of the n-th picture in coded order. The final arrival time for each picture is always the sum of the initial arrival time and the time required for the bits associated with that picture to enter the CPB:

$$t_{af}(n) = t_{ai}(n) + b(n) \div \text{bit\_rate}[k]. \quad \text{(C-2)}$$

For each subsequent picture n, the initial arrival time is equal to the final arrival time of picture $n-1$, unless that time precedes the earliest arrival time, computed by

$$t_{ai,earliest}(n) = t_c \cdot \text{cpb\_removal\_delay}(n) - (\text{initial\_cpb\_removal\_delay\_offset} \div 90{,}000) \quad \text{(C-3)}$$

as indicated in:

$$t_{ai}(n) = \max\{ t_{af}(n-1), t_{ai,earliest}(n) \} \quad \text{(C-4)}$$

See subclauses D.2.1 and D.3.1 on the picture timing SEI Message for the syntax and semantics of cpb_removal_delay and initial_cpb_removal_delay_offset. When the encoder is producing a bit rate lower than the bit rate associated with a CPB, this rule may delay the entry of some pictures into the CPB, producing periods during which no data enters.

C.1.1.2 Timing of coded picture removal

For the first picture after receiving the buffering period SEI message associated with the random access point, the coded data associated with the picture is removed from the CPB at a *removal time* equal to the following:

$$t_r(0) = \text{initial\_cpb\_removal\_delay} \div 90000 \quad \text{(C-5)}$$

where initial_cpb_removal_delay is defined in the buffering period SEI message.

After the first picture is removed, the buffer is examined at subsequent points of time, each of which is delayed from the previous one by an integer multiple of the clock tick t_c.

The removal time $t_r(n)$ of coded data for picture n is delayed with respect to that of picture 0; the delay is equal to the time indicated in the cpb_removal_delay syntax element present in the picture t iming SEI Message.

$$t_r(n) = t_r(0) + t_c \cdot \text{cpb\_removal\_delay}(n) \quad \text{(C-6)}$$

At this time, the coded data for the next transmitted picture is removed from the CPB.

If low_delay_hrd_flag is equal to 1, and if the amount of coded data for picture n, $b(n)$, is so large that it prevents removal at the computed removal time, the coded data is removed at the *delayed removal time*, $t_{r,d}(n, m^*)$, given by

$$t_{r,d}(n,m^*) = t_r(0) + t_c \cdot m^*, \quad \text{(C-7)}$$

where m^* is such that $t_{r,d}(n, m^*-1) < t_{af}(n) \le t_{r,d}(n, m^*)$. This is an aspect of low-delay operation (see subclause C.2.1.2). This delayed removal time is the next time instant after the final arrival time $t_{af}(n)$ which is delayed with respect to $t_r(0)$ by an integer multiple of t_c.

C.1.1.3 Conformance constraints on coded bitstreams

A transmitted or stored stream of coded data conforming to this Recommendation | International Standard fulfils the following requirements.

- ***Removal time consistency.*** For each picture, the removal ti mes $t_r(n)$ computed using different buffering periods as starting points for conformance verification shall be consistent to within the precision of the clocks used (90 kHz clock used for initial removal time, t_c clock used for subsequent removal time calcu lations, and bit_rate[k] used for arrival times).

Note - This can be ensured at the encoder by computing the initial CPB removal delay for a buffering period SEI message from the arrival and removal times computed using Equations C-4 and C-6. That is, if the last picture before the random access point is picture $n-1$, then the initial CPB removal delay for the next buffering period SEI message could be computed by

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

$$initial\_cpb\_removal\_delay = 90000 \times (t_r(n) - t_r(n-1) + t_{ai}(n)) \quad \text{(C-8)}$$

- ***Underflow and Overflow Prevention.*** The buffer shall never overflow or underflow.

NOTE - In terms of the arrival and removal schedules, this means that, with the exception of some pictures in low-delay mode that are described below, all bits from a picture must be in the CPB at the picture's computed removal time $t_r(n)$. In other words, its final arrival time must be no later than its removal time: $t_{af}(n) \leq t_r(n)$. Further, the removal time $t_r(n)$ must be no later than the time-equivalent of the buffer size te[cpb_size[k]]. Note that this prevents both underflow and overflow.

- ***Big Picture Removal Time, Overflow Prevention and Resynchronisation of Underflow Prevention.*** If the final arrival time $t_{af}(n)$ of picture n exceeds its computed removal time $t_r(n)$, its size must be such that it can be removed from the buffer without overflow at $t_{r,d}(n,m)$ as specified above.

- ***Constant Bit Rate Constraint.*** If vbr_cbr_flag[k] = = 1, data shall arrive continuously at the input to the CPB. This is equivalent to ensuring that

$$t_{af}(n-1) \geq t_{ai,earliest}(n) \quad \text{(C-9)}$$

- ***Maximum Removal Rate.*** The difference between consecutive removal times, $t_r(n) - t_r(n-1)$, shall not be less than the ratio of the number of macroblocks in picture n to the maximum sample processing rate for the bitstream's level in macroblocks per second (see subclause A.3.1).

C.1.2 Operation of the decoded picture buffer (DPB) verifier

[Ed.Note:

- The description in Annex A should be much clearer about the relationship between the reference memory numbers in Table A-1 and the actual number of frames allowed for the Decoded Picture Buffer. And the term "reference memory" in Table A-1 should be changed to "Decoded Picture Buffer Memory."
- The DPB needs availiblity of DTS and OTS values and the operations of marking and unmarking defined in clause 8.
- Primary vs. redundant pictures (argh!)]

The decoded picture buffer contains frame stores. These frame stores may contain frames and/or fields that are to be used for reference in future decoding (reference pictures) and/or are held for future output (reordered and delayed pictures). At $t_r(0)$, the DPB fullness, measured in frames, is initialized to 0. The following four steps all happen instantaneously at $t_r(n)$, the CPB removal time of picture n, and in the sequence listed.

C.1.2.1 Removal of pictures from the DPB

[Ed.Note. Figure C-2 needs changed to reflect the new design.]

A picture's DPB output time $t_{o,dpb}(n)$ is computed by

$$t_{o,dpb}(n) = t_r(n) + t_c \cdot \text{dpb\_output\_delay}(n) \quad \text{(C-10)}$$

First, some pictures may be removed from the buffer. Picture m is removed if it satisfies both of the following rules:

- It has been marked as "unused for reference" during the decoding process of previously decoded pictures.
- Its DPB output time is less than or equal to the CPB removal time of the current picture; *i.e.* $t_{o,dpb}(m) \leq t_r(n)$

If picture n is an IDR picture additional pictures are removed from the DPB. If non_ref_pic_reset_flag is equal to 0, all reference pictures are removed from the DPB. If non_ref_pic_reset_flag is equal to 1, all pictures are removed from the DPB. Following the removal of any frame or the removal of the last field in a frame store, the fullness is decremented by one.

The ordering of DPB output times for the sequence of decoded pictures is required to be the same as the ordering of the picture order counts for the sequence of decoded pictures (subclause 8.2.2).

C.1.2.2 Picture decoding

After the removal step, picture n is decoded.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

C.1.2.2.1 Storage of Reference Pictures into the DPB

Picture *n* is added to the decoded picture buffer if it is marked as "used for reference" (through the nal_ref_idc). If the picture is a frame, it consumes an entire frame store. If it is a coded field and if there is no field of the opposite parity with the same frame number already in the DPB, it consumes an entire frame store. Otherwise, it consumes no additional memory.

C.1.2.2.2 Storage of Non-reference Pictures into the DPB

If the DPB output time of picture *n* exceeds its CPB removal time, it is stored in the DPB. If the picture is a frame, the fullness of the DPB is incremented by one. If it is a coded field and if there is no non -reference field of the opposite parity already in the DPB, the fullness of the DPB is incremented by one. Otherwise, it is paired with the non -reference field of the opposite parity which has the earliest DPB output time.

C.1.2.3 Unused picture marking

Next, mark each picture in the decoded picture buffer as "unused for reference" if it is no longer needed for reference. This may arise through an MMCO command associated with picture *n* (see subclauses 8.3.6.7.2, 8.3.6.7.3, 8.3.6.7.5, 8.3.6.7.6 and 8.3.6.7.7) or through sliding wi ndow operation (see subclause 8.3.6.6).

C.1.2.4 Conformance constraints

- ***Overflow Prevention.*** Immediately after any picture is added to the DPB, the fullness of the DPB shall be less than or equal to the level limit on the DPB pictures (see Annex A subclauses on Le vel Limits for each profile, and subclause A.3.1).
- ***Maximum Output Rate.*** The difference between the output time of a picture and that of the picture output immediately preceding it shall be larger than the ratio of the number of macroblocks in the picture to the maximum sample processing rate for the bitstream's level in macroblocks per second (see subclause A.3.1).

[Ed. Note: (ev) this output rate restriction only affects, and should only affect, the values of the dpb_output_delay, and does not relate to the number of pictures "output" by MMCO commands. (A single MMCO command might mark a large number of picture unused.)]

[Ed. Note: Note that this is an output rate restriction, distinct from the deco ding rate restriction. We intended to do both, right?]

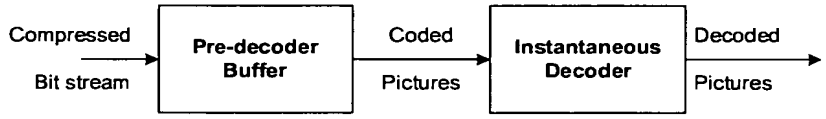
[Ed. Note: A conforming stream can place reordered frames into space for sliding window as long as the sliding window is not using it. Non-reference frames can be dropped without impacting the operat ion of the sliding window remaining conforming.]

C.2 Informative description of the HRD

This subclause does not form an integral part of this Recommendation | International Standard.

Subclause 0 contains the normative requirements i mposed by a set of buffering verifiers. This subclause provides explanatory text describing in more detail the operation and capabilities of these buffers.

The HRD Coded Picture Buffer represents a means to communicate how the bit rate is controlled in th e process of compression. The HRD contains a CPB through which compressed data flows with a precisely specified arrival and removal timing, as shown in Figure C-2. An HRD may be designed for variable or constant bit rate operatio n, and for low-delay or delay-tolerant behavior. The HRD described in this document handles all cases.



Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Figure C-3 – A Hypothetical Reference Decoder

Compressed data representing a sequence of *coded pictures* flows into the CPB according to a specified *arrival schedule*. All compressed bits associated with a given coded picture are removed from the CPB by the instantaneous decoder at the specified *removal time* of the picture.

The CPB *overflows* if the buffer becomes full and more bits are arriving. The buffer *underflows* if the removal time for a picture occurs before all compressed bits representing the picture have arrived.

C.2.1 Constrained arrival time leaky bucket (CAT-LB) model

The hypothetical reference decoder (HRD) is a mathematical model of a decoder and its input buffer. The k-th CPB of the HRD is characterized by the pre-decoder peak rate bit_rate[k] (in bits per second), the buffer size cpb_size[k] (in bits), the sequence initial CPB removal delay (in seconds), as well as picture removal delays for each picture. The first three of these parameters represent levels of resources (transmission capacity, buffer capacity, and delay) used to decode a bitstream.

The term "leaky bucket" arises from the analogy of the encoder as a system that "dumps" water in discrete chunks into a bucket that has a hole in it. The departure of bits from the encoder buffer corresponds to water leaking out of the bucket. Here, the decoder buffer is described, which has an inverse behaviour where bits flow *in* at a constant rate, and are removed in chunks.

The leaky bucket described here is called a *constrained arrival time* leaky bucket because the arrival times of all pictures after the first are constrained to arrive at the buffer input no earlier than the difference in hypothetical encoder processing times between that picture and the first picture. Furthermore, the arrival of all pictures in the CPB can be adjusted with a flexible offset. In other words, if a picture is encoded exactly seven seconds after the first picture was encoded and the offset is 0.5 seconds, then its bits are guaranteed not to start arriving in the buffer prior to 6.5 seconds after the bits of the first picture started arriving. It is possible to know this encoding time difference because the inter-picture delay is sent in the bitstream as the CPB removal delay and the offset is sent as the CPB initial removal delay offset.

C.2.1.1 Operation of the CAT-LB HRD

The HRD input buffer has capacity cpb_size[k] bits. Initially, the buffer begins empty. The lifetime in the buffer of the coded bits associated with picture n is characterized by the *arrival interval* $\{t_{ai}(n), t_{af}(n)\}$ and the *removal time* $t_r(n)$. The end-points of the arrival interval are known as the *initial arrival time* and the *final arrival time*.

At time $t_{ai}(0) = 0$, the buffer begins to receive bits at the rate bit_rate[k]. The removal time $t_r(0)$ for the first picture is computed from the pre-decoder removal delay initial_cpb_removal_delay (see Buffering Period SEI Message) associated with the buffer by the following:

$$t_r(0) = \text{initial\_cpb\_removal\_delay} \div 90\,000. \quad \text{(C-11)}$$

Removal times $t_r(1)$, $t_r(2)$, $t_r(3)$, ..., for subsequent pictures (in transmitted order) are computed with respect to $t_r(0)$, as follows. Let the *clock tick* t_c be specified by

$$t_c = \text{num\_units\_in\_tick} \div \text{time\_scale} \quad \text{(C-12)}$$

For instance, if time_scale = 60 000 and num_units_in_tick = 1 001, then

$$t_c = 1\,001 \div 60\,000 = 16.68333\ldots \text{ milliseconds.} \quad \text{(C-13)}$$

In the picture piming SEI message for each picture, there is a cpb_removal_delay syntax element. This indicates the number of clock ticks to delay the removal of picture n after removing picture 0. Thus, the removal time is simply

$$t_r(n) = t_r(0) + t_c \times \text{cpb\_removal\_delay}(n) \quad \text{(C-14)}$$

The calculation of arrival times is more complex, because of the arrival time constraint. The initial arrival time of picture n is equal to the final arrival time of picture $n-1$, unless that time precedes the earliest arrival time, computed by

$$t_{ai,earliest}(n) = t_c \times \text{cpb\_removal\_delay}(n) - (\text{initial\_cpb\_removal\_delay\_offset} \div 90\,000) \quad \text{(C-15)}$$

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Let $b(n)$ be the number of bits associated with picture n. The duration of the picture arrival interval is always the time - equivalent of the picture size in bit s, at the rate bit_rate[k].

$$t_{af}(n) - t_{ai}(n) \equiv te[b(n)] = b(n) \div \text{bit\_rate[k]} \quad \text{(C-16)}$$

Figure C-3 demonstrates a segment of the CPB fullness plot for a CAT-LB with the parameters given in Table C-1 and picture sizes given by the first column of Table C-2. Note that Table C-2 lists for each picture the values for many times of interest in the buffering process. In addition to quantiti es specified above, the second column of Table C-2 contains t_e, which represents a hypothetical encoding time equal to the earliest possible initial arrival time of the picture.

Table C-1 - Attributes of an example CAT-LB HRD

Attribute	Value	Units
time_scale	1	units per second
num_units_in_tick	1	units per tick
bit_rate	1 000	bits per second
cpb_size	10 000	bits
initial_delay	10	seconds
initial_delay_offset	0	seconds

Table C-2 - Picture sizes, and encoding, arrival and removal times for the example CAT -LB HRD

b	t_e	t_{ai}	t_{af}	$t_{ai}-t_e$	t_r	$t_r - t_{ai}$	$t_r - t_e$
5 000	0	0	5	0	10	10	10
1 000	1	5	6	4	11	6	10
1 000	2	6	7	4	12	6	10
1 000	3	7	8	4	13	6	10
1 000	4	8	9	4	14	6	10
1 000	5	9	10	4	15	6	10
500	6	10	10.5	4	16	6	10
500	7	10.5	11	3.5	17	6.5	10
500	8	11	11.5	3	18	7	10
500	9	11.5	12	2.5	19	7.5	10
500	10	12	12.5	2	20	8	10
500	11	12.5	13	1.5	21	8.5	10
500	12	13	13.5	1	22	9	10
500	13	13.5	14	0.5	23	9.5	10
500	14	14	14.5	0	24	10	10
500	15	15	15.5	0	25	10	10
500	16	16	16.5	0	26	10	10
500	17	17	17.5	0	27	10	10
3 000	18	18	21	0	28	10	10
3 000	19	21	24	2	29	8	10
3 000	20	24	27	4	30	6	10
3 000	21	27	30	6	31	4	10
2 000	22	30	32	8	32	2	10
300	23	32	32.3	9	33	1	10
300	24	32.3	32.6	8.3	34	1.7	10
300	25	32.6	32.9	7.6	35	2.4	10
300	26	32.9	33.2	6.9	36	3.1	10

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

300	27	33.2	33.5	6.2	37	3.8	10
300	28	33.5	33.8	5.5	38	4.5	10
300	29	33.8	34.1	4.8	39	5.2	10
300	30	34.1	34.4	4.1	40	5.9	10
300	31	34.4	34.7	3.4	41	6.6	10
300	32	34.7	35	2.7	42	7.3	10
300	33	35	35.3	2	43	8	10
300	34	35.3	35.6	1.3	44	8.7	10
300	35	35.6	35.9	0.6	45	9.4	10
300	36	36	36.3	0	46	10	10
300	37	37	37.3	0	47	10	10
300	38	38	38.3	0	48	10	10
300	39	39	39.3	0	49	10	10
300	40	40	40.3	0	50	10	10
300	41	41	41.3	0	51	10	10
300	42	42	42.3	0	52	10	10
500	43	43	43.5	0	53	10	10
500	44	44	44.5	0	54	10	10
500	45	45	45.5	0	55	10	10
500	46	46	46.5	0	56	10	10
500	47	47	47.5	0	57	10	10
500	48	48	48.5	0	58	10	10
500	49	49	49.5	0	59	10	10
500	50	50	50.5	0	60	10	10
500	51	51	51.5	0	61	10	10
500	52	52	52.5	0	62	10	10

Legend:

t_e Encoding time

t_{ai} Initial arrival time

t_{af} Final arrival time

t_r Removal time

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

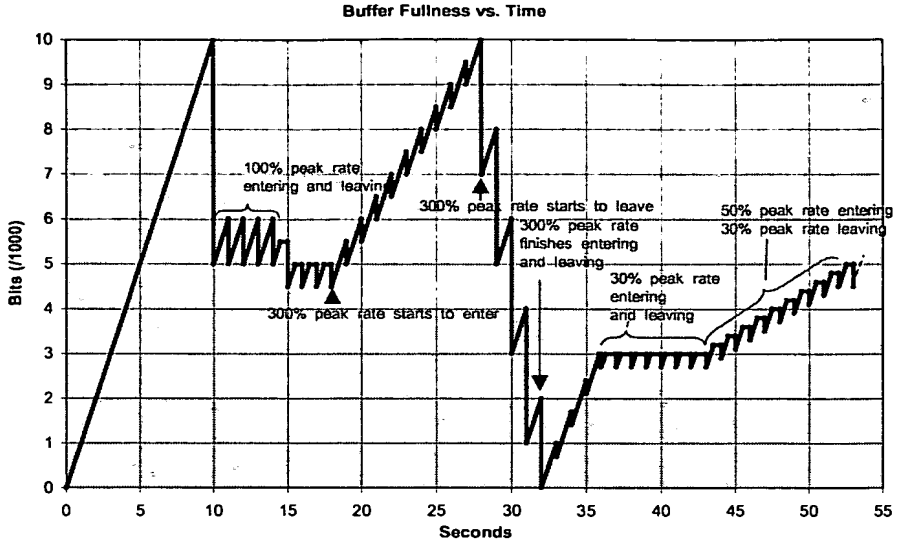


Figure C-4 – Buffer fullness plot for example HRD in Table C-2 with picture sizes given in Table C-3

As can be seen from Figure C-4, the initial picture is large, and is followed by five pictures at exactly the buffer arrival rate R. This is followed by twelve pictures at half the rate, four pictur es at three times the rate and one picture at twice the rate. Following this are two segments with pictures at 30% and 50% of the rate, respectively. In Figure C-4, the time interval from 10 seconds to 18 seconds illustrates the behaviour when the bit rate is constant and at or below the rate R. In fact, whenever the arrival bit rate remains less than R for a time, the lower points of the fullness curve will not change. It can be noted that data enters the buffer continuously at rate R until t=14.5 seconds. At that time, fifteen pictures have entered the buffer, and the sixteenth picture cannot enter because its earliest possible arrival time is t=15. Thus there is a 0.5 second flat spot from 14.5 to 15 seconds. The reason it t akes until t=14.5 for this to occur is that the initial picture took five picture periods to enter the buffer. Further, the fullness at the peak in such a segment will be proportional to the fraction of the peak rate being consumed by the pictures. From seconds 18 to 28, we see the temporary effect of an increase in arrival rate to above R. Once those large pictures start to exit the buffer, the bit rate of pictures leaving the buffer exceeds R, and the fullness decreases. This process terminates at sec ond 32, when the big pictures have exited and the series of smaller pictures starts entering the buffer. During seconds 36 -43, the 30% peak rate pictures are entering and leaving the buffer, and during seconds 43 -52, 30% peak rate pictures are leaving whi le 50% peal rate pictures are entering. Hence the buffer fullness rises. Once 50% peak rate pictures begin to leave, the fullness stabilizes at 50% full. Note that this CPB stabilizes at a fullness that is proportional to the ratio of the short -term average bit rate to the arrival bit rate, rather than at 100%.

In general, the curve of buffer fullness vs. time is given by the following expression:

$$BF(t) = \sum_{n} \left[I(t_{af}(n) \le t < t_r(n)) \times b(n) + I(t_{ai}(n) < t < t_{af}(n)) \times be(t - t_{ai}(n)) \right] \qquad \text{(C-17)}$$

This expression uses indicator functions $I(\,)$ with time-related logical assertions as arguments to sum only those pictures that are completely in the buffer at time t, plus the appropriate portion of the picture currently entering the buffer, if one is. The indicator function $I(x)$ is equal to 1 if x is true and equal to 0 otherwise.

C.2.1.2 Low-delay operation

Low-delay operation of the CPB is obtained by selecting a low value for the initial pre -decoder removal delay. This results in true low delay through the buffer because, under normal operation, no removal delay ($t_r(n)$-$t_{ai}(n)$) can exceed the initial removal delay minus the initial removal delay offset. To see this, consider that the maximum removal delay

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

for picture n occurs when the initial arrival time is equal to the earliest arrival time. Therefore, the maximum removal delay is given by $t_r(n) - t_{ai,earliest}(n)$. But,

$$t_r(n) = t_r(0) + t_c \times \text{cpb\_removal\_delay}(n), \quad \text{(C-18)}$$

and

$$t_{ai,earliest}(n) = t_c \times \text{cpb\_removal\_delay}(n) - (\text{initial\_cpb\_removal\_delay\_offset} \div 90\,000), \quad \text{(C-19)}$$

so

$$t_r(n) - t_{ai,earliest}(n) = t_r(0) + (\text{initial\_cpb\_removal\_delay\_offset} \div 90\,000). \quad \text{(C-20)}$$

Thus setting an initial low delay creates a steady-state low-delay condition. Note that the initial CPB removal delay is expected to be set to 0 or a very low value in the low delay mode.

However, in low-delay operation, it is useful to be able to process the occasional large picture whose size is so large than that it cannot be removed by its indicated removal time. Such a large picture can arise at a scene change, for example. This would ordinarily lead to an "underflow" condition. When a large picture is encountered, the rules for removal are relaxed to prevent this. The picture is removed at the *delayed removal time*, $t_{r,d}(n, m^*)$, given by

$$t_{r,d}(n,m^*) = t_r(0) + t_c \times m^* + (\text{initial\_cpb\_removal\_delay\_offset} \div 90\,000), \quad \text{(C-21)}$$

where m^* is such that $t_{r,d}(n, m^*-1) < t_{ai}(n) + te[b(n)] \le t_{r,d}(n, m^*)$. Note that the buffer must be large enough that this large picture can be accommodated without overflow. Immediately after such a picture is received the removal time of the next picture must be such that low-delay operation is resumed. An encoder can facilitate this by skipping a number of pictures immediately after the large picture, if necessary.

C.2.1.3 Bitstream / packet stream constraints

The buffer must not be allowed to underflow or overflow. Furthermore, all pictures except the isolated big pictures must be completely in the buffer before their computed removal times. Isolated big pictures are allowed to arrive later than their computed removal times, but must still obey the overflow constraint. In CBR mode, there must be no gaps in bit arrival.

C.2.1.3.1 Underflow

The underflow constraint, $BF(t) \ge 0$ for all t, is satisfied if the final arrival time of each picture precedes its removal time.

$$t_{af}(n) \le t_r(n) \quad \text{(C-22)}$$

This puts an upper bound on the size of picture n. The picture size can be no larger than the bit-equivalent of the time interval from the start of arrival to the removal time.

$$b(n) \le be[t_r(n) - t_{ai}(n)] \quad \text{(C-23)}$$

Since the initial arrival time $t_{ai}(n)$ is in general a function of the sizes and removal delays of previous pictures, the constraint on $b(n)$ will vary over time as well.

C.2.1.3.2 Overflow

Overflow is avoided provided the buffer fullness curve $BF(t)$ never exceeds the buffer size B.

The constraints that the initial pre-decoder removal delay must be no larger than the time-equivalent of the buffer size, $t_r(0) \le te(B)$, and that under normal operation no removal delay can exceed the initial one plus the initial offset guarantee that no overflow occurs in normal operation. To avoid overflow of an isolated big picture, the picture size is constrained by

$$b(n) \le be[B - t_{ai}(n)] \quad \text{(C-24)}$$

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

C.2.1.3.3 Constant bitrate (CBR) operation

The CAT-LB model operates in constant bit rate mode if one further constraint is applied - that data must constantly arrive at the input of the buffer. This ensures that the average rate is equal to the buffer rate R. This model behaves like an ISO/IEC 11172-2 (MPEG-1) CBR model with variable frame rate. This condition is ensured if the final arrival time of picture $n-1$ is no earlier than the earliest initial arrival time of picture n.

$$t_{af}(n-1) \geq t_{ai,earliest}(n) \qquad \text{(C-25)}$$

This time constraint puts a lower bound on $b(n)$.

NOTE: A bitstream that satisfies an ISO/IEC 11172-2 or an ITU-T Recommendation H.262 | ISO/IEC 13818-2 CBR video buffering verifier at a given buffer size, initial delay and bitrate will also satisfy a CAT-LB with the same parameters.

C.2.1.4 Rate control considerations

An encoder employs rate control as a means to constrain the varying bit rate characteristics of the coded bitstream or packet stream in order to produce high quality coded pictures at the target bit rate(s). A rate control algorithm may target a variable bit rate (VBR) or a constant bit rate (CBR). It may even target both a high peak rate using a VBR scheme and an average rate using a CBR scheme. Further, as shown in subclause **Error! Reference source not found.**, multiple VBR rates can be targeted.

Rate control must ensure conformance with the CPB. This is related to the first goal of rate control, but is not necessarily the same. In this subclause, the way the CPB influences rate control is discussed. In a VBR CPB, the buffer must not overflow or underflow, but gaps may appear in the arrival rate. In order to meet these constraints, the encoder must ensure that for all t, the following inequalities remain true:

$$0 \leq BF(t) \leq B, \text{ for all } t. \qquad \text{(C-26)}$$

Using Equation D-14, this becomes:

$$0 \leq \sum_n \left[I(t_{af}(n) \leq t < t_r(n)) \times b(n) + I(t_{ai}(n) < t < t_{af}(n)) \times be(t - t_{ai}(n)) \right] \leq B, \text{ for all } t. \qquad \text{(C-27)}$$

The buffer fullness $B(t)$ is a piecewise non-decreasing function of time, with each non-decreasing interval bounded by two consecutive removal times. Therefore, it is sufficient to guarantee the conformance a t the interval endpoints; *i.e.* at the removal times. In particular, if underflow is prevented at the start of an interval (just after removal of a picture), it is completely prevented. The same holds for overflow at the end of the interval, just prior to picture removal. Therefore, the points of interest are the removal times. In the buffer at $t_r^-(n)$ and contributing to Equation C.-22 are picture n and possibly some additional pictures up to picture $m>n$ (with the last picture possibly only partially in the buffer). All pictures earlier than picture n have been removed. At $t_r^+(n)$, picture n has been removed.

Thus when encoding picture n, one rate control task is to allocate bits to picture n and the others in the immediate future in such a way that overflow is prevented at $t_r^-(n)$, and underflow is prevented at $t_r^+(n)$. Most immediately, as long as $b(n)$ is small enough so that $te[b(n)] \leq t_r(n) - t_{ai}(n)$, both overflow at $t_r^-(n)$ and underflow at $t_r^+(n)$ are prevented. This is usually a very high limit, and a rate control method will most likely further limit $b(n)$ through its bit allocation process.

C.2.2 Multiple leaky bucket description

C.2.2.1 Schedule of a bitstream

The sequence of removal time and picture size pairs $\{(t_r(n), b(n)), n=0,1,\ldots\}$ is called the schedule of a bitstream. The schedule of a bitstream is intrinsic to the bitstream, and completely characterizes the instantaneous coding rate of the bitstream over its lifetime. Although a bitstream may conform to CPBs with different peak bit rates and different p re-decoded buffer sizes, the schedule of the bitstream is independent of the CPB.

C.2.2.2 Containment in a leaky bucket

A leaky bucket with leak rate R_1, bucket size B_1, and initial bucket fullness B_1-F_1 is said to contain a bitstream with schedule $\{(t_r(n), b(n)), n=0,1,\ldots\}$ if the bucket does not overflow under the following conditions. At t_0, d_0 bits are inserted into the leaky bucket on top of the B_1-F_1 bits already in the bucket, and the bucket begins to drain at rate R1 bits per second. If the bucket empties, it remains empty until the next insertion. At time t_i, $i \geq 1$, d_i bits are inserted into the bucket, and the bucket continues to drain at rate R_1 bits per second. In other words, for $i \geq 0$, the state of the bucket just prior to time t_i is

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

$$b_0 = B_1 - F_1 \tag{C-28}$$

$$b_{i+1} = \max\{0, b_i + d_i - R_1(t_{i+1} - t_i)\}. \tag{C-29}$$

The leaky bucket does not overflow if $b_i + d_i \le B_1$ for all $i \ge 0$.

Equivalently, the leaky bucket contains the bitstream if the graph of the schedule o f the bitstream lies between two parallel lines with slope R_1, separated vertically by B_1 bits, possibly sheared horizontally, such that the upper line begins at F_1 at time t_0, as illustrated in Figure C-5. Note from Figure C-5 that the same bitstream is containable in more than one leaky bucket. Indeed, a bitstream is containable in an infinite number of leaky buckets.

Figure C-5 – Illustration of the leaky bucket concept
[Ed. Note: Something wrong with figure]

If a bitstream is contained in a leaky bucket with parameters (R_1,B_1,F_1), then when it is input with peak rate R_1 to a hypothetical reference decoder with parameters R=R_1, B=B_1, and F=F_1, then the HRD buffer does not overflow or underflow.

C.2.2.3 Minimum buffer size and minimum peak rate

If a bitstream is contained in two leaky buckets with parameters (R_1,B_1,F_1) and (R_2,B_2,F_2), then it is also contained in any leaky bucket with parameters (R,B,F) where a) $R_1 \le R \le R_2$, b) $B \ge B_{min}(R)$, and c) $F \ge F_{min}(R)$ and $B_{min}(R)$ and $F_{min}(R)$ are specified by

$$B_{min}(R) = \alpha B_n + (1 - \alpha)B_{n+1}, \tag{C-30}$$

$$F_{min}(R) = \alpha F_n + (1 - \alpha)F_{n+1}, \tag{C-31}$$

and

$$\alpha = (R_{n+1} - R) \div (R_{n+1} - R_n). \tag{C-32}$$

For $R \le R_1$,

$$B_{min}(R) = B_1 + (R_1 - R)T \tag{C-33}$$

$$F_{min}(R) = F_1, \tag{C-34}$$

where $T = t_{L-1} - t_0$ is the duration of the bitstream (i.e., the difference between the dec oding times for the first and last pictures in the bitstream). And for $R \ge R_N$,

$$B_{min}(R) = B_N \tag{C-35}$$

$$F_{min}(R) = F_N. \tag{C-36}$$

Thus, the leaky bucket parameters can be linearly interpolated and extrapolated.

Alternatively, when the bitstream is communicated to a decoder with bu ffer size B, it is decodable provided $\ge R_{min}(B)$ and $F \ge F_{min}(B)$, where for $B_n \ge B \ge B_{n+1}$,

$$R_{min}(B) = \alpha R_n + (1 - \alpha)R_{n+1} \tag{C-37}$$

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

$$F_{min}(B) = \alpha F_n + (1 - \alpha)F_{n+1} \quad \text{(C-38)}$$

$$\alpha = (B - B_{n+1}) \div (B_n - B_{n+1}). \quad \text{(C-39)}$$

For $B \geq B_1$,

$$R_{min}(B) = R_1 - (B - B_1) \div T \quad \text{(C-40)}$$

$$F_{min}(B) = F_1. \quad \text{(C-41)}$$

For $B \leq B_N$, the stream may not be decodable.

In summary, the bitstream is guaranteed to be decodable in the sense that the HRD buffer does not overflow or underflow, provided that the point (R,B) lies on or above the lower convex hull of the set of points (0, B_1+R_1T), (R_1,B_1), ..., (R_N,B_N), as illustrated in Figure C-6. The minimum start-up delay necessary to maintain this guarantee is $F_{min}(R) \div R$.

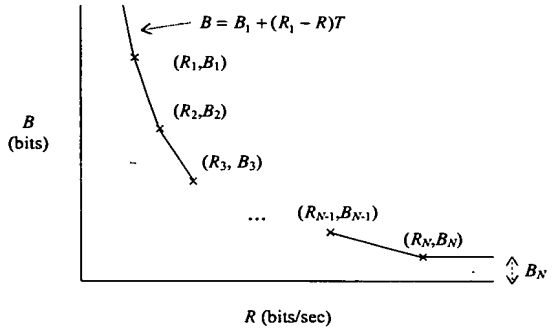


Figure C-6 – Further illustration of the leaky bucket concept

An HRD with buffer size B and initial decoder buffer fullness F with peak input rate R shall perform the tests $B \geq B_{min}(R)$ and $F \geq F_{min}(R)$, as specified above, for any conforming bitstream with LB parameters (R_1,B_1,F_1),...,(R_N,B_N,F_N), and shall decode the bitstream provided that $B \geq B_{min}(R)$ and $F \geq F_{min}(R)$.

C.2.2.4 Encoder considerations

The encoder can create a bitstream that is contained by some given N leaky buckets, or it can simply compute N sets of leaky bucket parameters after the bitstream is generated, or a combination of these. In the former, the encoder enforces the N leaky bucket constraints during rate control. Conventional rate control algorithm s enforce only a single leaky bucket constraint. A rate control algorithm that simultaneously enforces N leaky bucket constraints can be obtained by running a conventional rate control algorithm for each of the N leaky bucket constraints, and using as the current quantisation parameter (QP) the maximum of the QP's recommended by the N rate control algorithms.

Additional sets of leaky bucket parameters can always be computed after the fact (whether rate controlled or not), from the bitstream schedule for any given R_n, from the iteration specified in subclause C.2.2.2.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Annex D
Supplemental enhancement information

(This annex forms an integral part of this Recommendation | International Standard)

This annex specifies syntax and semantic for supplemental enhancement information that provides data constructs that are synchronous with the video data content. Each sei_payload( ) specifies PayloadType and PayloadSize parameters. This annex specifies supplemental enhancement information (SEI) that provides a data delivery mechanism construct that is delivered synchronous with the video data content. SEI assists in the processes related to decoding or display of video. SEI is not required for reconstructing the luma or chroma samples by a video decoder, and decoders are not required to process this information for conformance to this Recommendation | International Standard.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

D.1 SEI payload syntax

sei_payload(PayloadType, PayloadSize) {	C	Descriptor
if(PayloadType = = 0)		
buffering_period(PayloadSize)	7	
else if(PayloadType = = 1)		
pic_timing(PayloadSize)	7	
else if(PayloadType = = 2)		
pan_scan_rect(PayloadSize)	7	
else if(PayloadType = = 3)		
filler_payload(PayloadSize)	7	
else if(PayloadType = = 4)		
user_data_registered_itu_t_t35(PayloadSize)	7	
else if(PayloadType = = 5)		
user_data_unregistered(PayloadSize)	7	
else if(PayloadType = = 6)		
random_access_point(PayloadSize)	7	
else if(PayloadType = = 7)		
dec_pic_buffer_management_repetiti on(PayloadSize)	7	
else if(PayloadType = = 8)		
spare_pic(PayloadSize)	7	
else if(PayloadType = = 9)		
scene_info(PayloadSize)	7	
else if(PayloadType = = 10)		
subseq_info(PayloadSize)	7	
else if(PayloadType = = 11)		
subseq_layer_characteristics(PayloadSize)	7	
else if(PayloadType = = 12)		
subseq_characteristics(PayloadSize)	7	
else if(PayloadType = = 13)		
full_frame_freeze(PayloadSize)	7	
else if(PayloadType = = 14)		
full_frame_freeze_release(PayloadSize)	7	
else if(PayloadType = = 15)		
full_frame_snapshot(PayloadSize)	7	
else if(PayloadType = = 16)		
progressive_refinement_segment_start(PayloadSize)	7	
else if(PayloadType = = 17)		
progressive_refinement_segment_end(PayloadSize)	7	
else		
reserved_sei_message(PayloadSize)	7	
if(!byte_aligned( )) {		
bit_equal_to_one /* equal to 1 */	7	f(1)
while(!byte_aligned( ))		
bit_equal_to_zero /* equal to 0 */	7	f(1)
}		
}		

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

D.1.1 Buffering period SEI message syntax

buffering_period(PayloadSize) {	C	Descriptor
seq_parameter_set_id	7	ue(v)
if(nal_hrd_ parameters_present_flag = = 1) {		
for(k = 0; k <= cpb_cnt; k++) {		
initial_cpb_removal_delay[k]	7	u(v)
initial_cpb_removal_delay_offset[k]	7	u(v)
}		
}		
if(vcl_hrd_ parameters_present_flag = = 1) {		
for(k = 0; k <= cpb_cnt; k++) {		
initial_cpb_removal_delay[k]	7	u(v)
initial_cpb_removal_delay_offset[k]	7	u(v)
}		
}		
}		

D.1.2 Picture timing SEI message syntax

pic_timing(PayloadSize) {	C	Descriptor
if(nal_hrd_ parameters_present_flag || vcl_hrd_ parameters_present_flag) {		
cpb_removal_delay	7	u(v)
dpb_output_delay	7	u(v)
}		
pic_structure_present_flag	7	u(1)
if(pic_structure_present_flag) {		
pic_structure	7	u(3)
for(i = 0 ; i < NumClockTS ; i++) {		
clock_timestamp_flag[i]	7	u(1)
if (clock_timestamp_flag[i]) {		
ct_type	7	u(2)
nuit_field_based_flag	7	u(1)
counting_type	7	u(5)
full_timestamp_flag	7	u(1)
discontinuity_flag	7	u(1)
cnt_dropped_flag	7	u(1)
nframes	7	u(8)
if(full_timestamp_flag) {		
seconds_value /* 0,…,59 */	7	u(6)
minutes_value /* 0,…,59 */	7	u(6)
hours_value /* 0,…,23 */	7	u(5)
} else {		
seconds_flag	7	u(1)
if(seconds_flag) {		

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

seconds_value /* range 0,...,59 */	7	u(6)
minutes_flag	7	u(1)
if(minutes_flag) {		
minutes_value /* 0,...,59 */	7	u(6)
hours_flag	7	u(1)
if(hours_flag)		
hours_value /* 0,...,23 */	7	u(5)
}		
}		
}		
if(time_offset_length > 0)		
time_offset	7	i(v)
}		
}		
}		

D.1.3 Pan-scan rectangle SEI message syntax

pan_scan_rect(PayloadSize) {	C	Descriptor
pan_scan_rect_id	7	ue(v)
pan_scan_rect_left_offset	7	se(v)
pan_scan_rect_right_offset	7	se(v)
pan_scan_rect_top_offset	7	se(v)
pan_scan_rect_bottom_offset	7	se(v)
}		

D.1.4 Filler payload SEI message syntax

filler_payload(PayloadSize) {	C	Descriptor
for(k = 0; k < PayloadSize; k++)		
filler_byte /* equal to 0xFF */	7	f(8)
}		

D.1.5 User data registered by ITU-T Recommendation T.35 SEI message syntax

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

user_data_registered_itu_t_t35(PayloadSize) {	C	Descriptor
itu_t_t35_country_code	7	b(8)
if(country_code != 0xFF)		
i = 1;		
else {		
itu_t_t35_country_code_extension_byte	7	b(8)
i = 2;		
}		
do {		
itu_t_t35_payload_byte	7	b(8)
i++		
} while(i < PayloadSize)		
}		

D.1.6 User data unregistered SEI message syntax

user_data_unregistered(PayloadSize) {	C	Descriptor
for(i = 0; i < PayloadSize; i++)		
user_data_arbitrary_payload_byte	7	b(8)
}		

D.1.7 Random access point SEI message syntax

random_access_point(PayloadSize) {	C	Descriptor
recovery_frame_cnt	7	ue(v)
exact_match_flag	7	u(1)
broken_link_flag	7	u(1)
}		

D.1.8 Decoded picture buffer management Repetition SEI message syntax

dec_pic_buffer_management_repetition(PayloadSize) {	C	Descriptor
original_frame_num	7	u(v)
dec_pic_buffer_management( )		
}		

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

D.1.9 Spare picture SEI message syntax

spare_pic(PayloadSize) {	C	Descriptor
target_frame_num	7	ue(v)
num_spare_pics_minus1	7	ue(v)
for(i = 0; i < num_spare_pics_minus1+1; i++) {		
delta_spare_frame_num	7	ue(v)
ref_area_indicator	7	ue(v)
if(ref_area_indicator == 1)		
for(j = 0; j < number_of_mbs_in_pic; j++)		
ref_mb_indicator	7	u(1)
else if(ref_area_indicator == 2) {		
MbCnt = 0		
do {		
zero_run_length	7	ue(v)
MbCnt = MbCnt + zero_run_length + 1		
} while(MbCnt <= MaxMbAddress)		
}		
}		
}		

D.1.10 Scene information SEI message syntax

[Ed.Note: function more_sei_payload_data( ) should be specified in an appropriate location, or sub -sequence-related SEI messages should have some syntax added to signal the presence of more data.]

scene_info(PayloadSize) {	C	Descriptor
scene_id	7	u(8)
scene_transition_type	7	ue(v)
if(scene_transition_type > 3)		
second_scene_id	7	u(8)
}		

D.1.11 Sub-sequence information SEI message syntax

subseq_info(PayloadSize) {	C	Descriptor
subseq_layer_num	7	ue(v)
subseq_id	7	ue(v)
first_ref_pic_flag	7	u(1)
leading_non_ref_pic_flag	7	u(1)
last_pic_flag	7	u(1)
if(more_sei_payload_data( ))		
subseq_frame_num	7	ue(v)
}		

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

D.1.12 Sub-sequence layer characteristics SEI message syntax

subseq_layer_characteristics(PayloadSize) {	C	Descriptor
do {		
accurate_statistics_flag	7	u(1)
average_bit_rate	7	u(16)
average_frame_rate	7	u(16)
} while(more_sei_payload_data( ))		
}		

D.1.13 Sub-sequence characteristics SEI message syntax

subseq_characteristics(PayloadSize) {	C	Descriptor
subseq_layer_num	7	ue(v)
subseq_id	7	ue(v)
duration_flag	7	u(1)
if(duration_flag)		
subseq_duration	7	u(32)
average_rate_flag	7	u(1)
if(average_rate_flag) {		
accurate_statistics_flag	7	u(1)
average_bit_rate	7	u(16)
average_frame_rate	7	u(16)
}		
num_referenced_subseqs	7	ue(v)
for(n = 0; n < num_referenced_subseqs; n++) {		
ref_subseq_layer_num	7	ue(v)
ref_subseq_id	7	ue(v)
}		
}		

D.1.14 Full-frame freeze SEI message syntax

full_frame_freeze(PayloadSize) {	C	Descriptor
}		

D.1.15 Full-frame freeze release SEI message syntax

full_frame_freeze_release(PayloadSize) {	C	Descriptor
}		

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

D.1.16 Full-frame snapshot SEI message syntax

full_frame_snapshot(PayloadSize) {	C	Descriptor
snapshot_id	7	u(32)
}		

D.1.17 Progressive refinement segment start SEI message syntax

progressive_refinement_segment_start(PayloadSize) {	C	Descriptor
progressive_refinement_id	7	u(32)
num_refinement_steps_minus1	7	ue(v)
}		

D.1.18 Progressive refinement segment end SEI message syntax

progressive_refinement_segment_end(PayloadSize) {	C	Descriptor
progressive_refinement_id	7	u(32)
}		

D.1.19 Reserved SEI message syntax

reserved_sei_message(PayloadSize) {	C	Descriptor
for(i=0; i<PayloadSize; i++)		
reserved_sei_message_payload_byte	7	b(8)
}		

D.2 SEI payload semantics

D.2.1 Buffering period SEI message semantics

A buffering period is specified as the set of pictures between two instances of the buffering period SEI message. The seq_parameter_set_id indicates the sequence p arameter set that contains the sequence HRD attributes.

seq_parameter_set_id indicates the sequence parameter set that contains the sequence HRD attributes.

initial_cpb_removal_delay: This syntax element represents the delay between the time of arrival i n the CPB of the first bit of the coded data associated with the first picture following the buffering period SEI message (including all NAL data in the case that the HRD pertains to the NAL) and the time of removal of the coded data associated with the sa me picture from the CPB. The syntax element is a fixed length code whose length in bits is given by initial_cpb_removal_delay_length_minus1+1. It is in units of a 90 kHz clock. The initial_cpb_removal_delay syntax element is used in conjunction with t he CPBs as specified in Annex C. initial_cpb_removal_delay shall not be equal to 0 and shall not exceed the time-equivalent of the CPB size.

initial_cpb_removal_delay_offset: This syntax element is used in combination with the cpb_removal_delay to comput e the initial arrival time of coded pictures in the CPB. It is in units of a 90 kHz clock. The initial_cpb_removal_delay_offset syntax element is a fixed length code whose length in bits is given by initial_cpb_removal_delay_length_minus1+1.

Over the entire sequence, the sum of initial_cpb_removal_delay and initial_cpb_removal_delay_offset shall be constant and shall not exceed the time-equivalent of the CPB size.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

D.2.2 Picture timing SEI message semantics

If the bitstream includes HRD data, then in the case that any HRD buffers are signalled in the sequence parameter set(s), the following statements are true.

1. At each random access point, a buffering period SEI message shall follow the random access point SEI message.
2. A picture timing SEI Message shall precede the first NAL Unit of each picture subsequent to the picture with the buffering period SEI message.

Each of these SEI messages pertains to the next picture that follows it in decoding order.

cpb_removal_delay: This syntax element indicates how many clock ticks (see Annex C) to wait after removal from the HRD CPB of the picture immediately following the most recent random access SEI message before removing from the buffer the picture data immediately following the SEI message which contains the element. This value is also used to calculate an earliest possible time of arrival of picture data into the CPB, as specified in Annex C. The syntax element is a fixed length code whose length in bits is given by cpb_removal_delay_length. The cpb_removal_delay is the remainder of a $2^{(cpb\_removal\_delay\_length\_minus1 + 1)}$ counter.

dpb_output_delay: This syntax element is used to compute the *DPB output time*. It indicates how many clock ticks to wait after removal of a picture from the CPB before it can be removed from the DPB (see Annex C).

NOTE - A picture is not removed from the DPB at its output time if it is still being used as a reference.

When field_pic_flag is equal to 1 and indicates multiple fields, the DPB output delay shall be interpreted to refer to the first field in time. [Ed. Note: needs improvement]

NOTE - The timing of other fields must be inferred from external information about the time interval between fields.

The size of the syntax element dpb_output_delay shall remain constant for the sequence and is given in bits by dpb_output_delay_length_minus1.

pic_structure_present_flag: If and only if this syntax element is equal to 1, the pic_structure syntax element is present and follows immediately.

pic_structure: This parameter indicates whether a picture should be displayed as a frame or one or more fields, according to Table D-1.

Table D-1 – Interpretation of pic_structure

Value	Indicated display of picture	Restrictions	NumClockTS
0	frame	field_pic_flag must be 0	1
1	top field	field_pic_flag must be 1, bottom_field_flag must be 0	1
2	bottom field	field_pic_flag must be 1, bottom_field_flag must be 1	1
3	top field, bottom field, in that order	field_pic_flag must be 0	2
4	bottom field, top field, in that order	field_pic_flag must be 0	2
5	top field, bottom field, top field repeated, in that order	field_pic_flag must be 0	3
6	bottom field, top field, bottom field repeated, in that order	field_pic_flag must be 0	3
7	reserved		

NumClockTS is determined by pic_structure as indicated in Table D-1. There are up to NumClockTS sets of clock timestamp information for a picture, as indicated by clock_timestamp[i] for each set. The sets of clock timestamp information apply to the field(s) or the frame associated with the picture by pic_structure. In the case of fields, the order of the timestamps is the same as the display order of the fields, as indicated by pic_structure. Two fields associated with a frame picture may have different values of ct_type. If EquivalentTimestamp is equal for two fields of opposite parity that are adjacent in display order, both with ct_type equal to 0 (progressive) or ct_type equalt to 2 (unknown), then the

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

two fields are indicated to have come fro m the same original frame. Two adjacent fields in display order shall have different values of EquivalentTimestamp if the value of ct_type for either field is 1 (interlaced).

The contents of the clock timestamp SEI message specify an ideal display time co mputed as

$$\text{EquivalentTimestamp} = ((HH * 60 + MM) * 60 + SS) * \text{time\_scale} + NF * (\text{num\_units\_in\_tick} * (1 + \text{nuit\_field\_based\_flag})) + TO, \quad \text{(D-1)}$$

in units of ticks of a clock with clock frequency equal to time_scale Hz. The E quivalentTimestamp is the ideal display time assuming a display that can display frames and fields at any time without regard to a frame or field rate. If two or more frames with pic_structure equal to 0 are adjacent in output order and have equal values of EquivalentTimestamp, the ideal display duration for all such frames except for the last frame in display order is 0, and ideally only the last frame is displayed. The EquivalentTimestamp is not used for HRD purposes. Note that nframes and NF are frame -based counts.

clock_timestamp_flag[i]: If this syntax element is equal to 1, a number of clock timestamp fields are present and follow immediately.

ct_type: This parameter indicates the type (interlaced or progressive) of the original material as follow s:

Table D-2 – Mapping of ct_type to original picture scan

Value	Original picture scan
0	progressive
1	interlaced
2	unknown
3	reserved

nuit_field_based_flag: Used in calculating EquivalentTimestamp, as specified in Equation D-1.

counting_type: Specifies the method of dropping values of the nframes parameter as specified in Table D-3.

Table D-3 – Definition of counting_type values

Value	Interpretation
0	no dropping of nframes count values and no use of time_offset
1	no dropping of nframes count values
2	dropping of individual zero values of nframes count
3	dropping of individual max_pps values of npictures count
4	dropping of the two lowest (value 0 and 1) nframes counts when seconds_value is zero and minutes_value is not an integer multiple of ten
5	dropping of unspecified individual nframes count values
6	dropping of unspecified numbers of unspecified nframes count values
7 - 31	reserved

full_timestamp_flag indicates whether the nframes parameter is followed by seconds_value or seconds_flag.

discontinuity_flag indicates whether the time difference between the current value of EquivalentTimestamp and the value of EquivalentTimestamp compu ted from the last previously clock timestamp in decoding order can be interpreted as a true time difference. A value of 0 indicates that the difference represents a true time difference.

cnt_dropped_flag indicates the skipping of one or more values of nfr ames using the counting method specified by counting_type.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

nframes indicates the value of *NF* used to compute the EquivalentTimestamp. nframes shall be less than

$$\text{MaxFPS} = \text{Ceil}\ (\ \text{time\_scale} \div \text{num\_units\_in\_tick}\ ) \qquad \text{(D- 2)}$$

If counting_type is equal to 2 and cnt_dropped_flag is equal to 1, nframes shall be equal to 1 and the value of nframes for the last previous picture in output order shall not be equal to 0 unless discontinuity_flag is equal to 1.

NOTE – When counting_type is equal to 2, the need for increasingly large magnitudes of *TO* in Equation D-1 when using fixed non-integer frame rates (e.g., 12.5 frames per second with time_scale equal to 25 and num_units_in_tick equal to 2 and nuit_field_based_flag equal to 0) can be avoided by occasionally skipping over the value nframes equal to 0 when counting (e.g., counting nframes from 0 to 12, then incrementing seconds_value and counting nframes from 1 to 12, then incrementing seconds_value and counting nframes from 0 to 12, etc.).

If counting_type is equal to 3 and cnt_dropped_flag is equal to 1, nframes shall be equal to 0 and the value of nframes for the last previous picture in output order shall not be equal to MaxFPS – 1 unless discontinuity_flag is equal to 1.

NOTE – When counting_type is equal to 3, the need for increasingly large magnitudes of *TO* in Equation D-1 when using fixed non-integer frame rates (e.g., 12.5 frames per second with time_scale equal to 25 and num_units_in_tick equal to 2 and nuit_field_based_flag equal to 0) can be avoided by occasionally skipping over the value nframes equal to MaxFPS when counting (e.g., counting nframes from 0 to 12, then incrementing seconds_value and counting nframes from 0 to 11, then incrementing seconds_value and counting nframes from 0 to 12, etc.).

If counting_type is equal to 4 and cnt_dropped_flag is equal to 1, nframes shall be equal to 2 and the indicated value of *SS* shall be zero and the indicated value of *MM* shall not be an integer multiple of ten and nframes for the last previous picture in output order shall not be equal to 0 or 1 unless discontinuity_flag is equal to 1.

NOTE – When counting_type is equal to 4, the need for increasingly large magnitudes of *TO* in Equation D-1 when using fixed non-integer frame rates (e.g., 30000÷1001 frames per second with time_scale equal to 60 000 and num_units_in_tick equal to 1 001 and nuit_field_based_flag equal to 1) can be reduced by occasionally skipping over the value nframes equal to MaxFPS when counting (e.g., counting nframes from 0 to 29, then incrementing seconds_value and counting nframes from 0 to 29, etc., until the seconds_value is zero and minutes_value is not an integer multiple of ten, then counting nframes from 2 to 29, then incrementing seconds_value and counting nframes from 0 to 29, etc.). This counting method is well known in industry and is often referred to as "NTSC drop-frame" counting.

If counting_type is equal to 5 or 6 and cnt_dropped_flag is equal to 1, nframes shall not be equal to 1 plus the value of nframes for the last previous picture in output order modulo MaxFPS unless discontinuity_flag is equal to 1.

NOTE – When counting_type is equal to 5 or 6, the need for increasingly large magnitudes of *TO* in Equation D-1 when using fixed non-integer frame rates can be avoided by occasionally skipping over some values of nframes when counting. The specific values of nframes that are skipped are not specified when counting_type is equal to 5 or 6.

seconds_flag indicates whether seconds_value is present when full_timestamp_flag is equal to 0.

seconds_value indicates the value of *SS* used to compute the EquivalentTimestamp. seconds_v alue shall not exceed 59. If not present, the last previous seconds_value in decoding order shall be used as *SS* to compute the EquivalentTimestamp.

minutes_flag indicates whether seconds_value is present when full_timestamp_flag is equal to 0 and seconds_ flag is equal to 1.

minutes_value indicates the value of *MM* used to compute the EquivalentTimestamp. minutes_value shall not exceed 59. If not present, the last previous minutes_value in decoding order shall be used as *MM* to compute the EquivalentTimesta mp.

hours_flag indicates whether seconds_value is present when full_timestamp_flag is equal to 0 and seconds_flag is equal to 1 and minutes_flag is equal to 1.

hours_value indicates the value of *HH* used to compute the EquivalentTimestamp. hours_value shal l not exceed 23. If not present, the last previousl hours_value in decoding order shall be used as *HH* to compute the EquivalentTimestamp.

bit_equal_to_one is a single bit which shall be equal to 1.

time_offset indicates the value of *TO* used to compute the EquivalentTimestamp. The number of bits used to represent time_offset shall be equal to time_offset_length. If time_offset is not present, the value 0 shall be used as *TO* to compute the EquivalentTimestamp.

D.2.3 Pan-scan rectangle SEI message semantics

The pan-scan rectangle SEI message parameters define the coordinates of a rectangle relative to the cropping rectangle of the picture parameter set. Each coordinate of this rectangle is specified in units of 1/16 th sample spacing relative to the luma sampling grid.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

pan_scan_rect_id contains an identifying number which may be used to identify the purpose of the pan -scan rectangle (for example, to identify the rectangle as the area to be shown on a particular display device or as the area that contains a particular actor in the scene). pan_scan_rect_id shall not exceed 0xFFFFFFFF.

Values of pan_scan_rect_id from 0 to 0xFF and from 0x200 to 0x7FFFFFFF may be used as specified externally. Values of pan_scan_rect_id from 0x100 to 0x1FF and from 0x80000000 to 0xFFFF FFFF are reserved for future use by ITU-T | ISO/IEC. Decoders encountering a value of pan_scan_rect_id in the range of 0x100 to 0x1FF or in the range of 0x80000000 to 0xFFFFFFFF shall ignore (remove from the bitstream and discard) it.

pan_scan_rect_left_offset, pan_scan_rect_right_offset, pan_scan_rect_top_offset, and **pan_scan_rect_bottom_offset** specify, as signed integer quantities in units of $1/16^{th}$ sample spacing relative to the luma sampling grid, the location of the pan -scan rectangle. The values of each of these four syntax elements shall be inside the interval from -2^{31} to $2^{31}-1$, inclusive.

The pan-scan rectangle is specified, in units of $1/16^{th}$ sample spacing relative to the luma sampling grid, as the area of the rectangle with horizontal coordinates from 16 * cropping_rect_left + pan_scan_rect_left_offset to 16 * [16 * PicWidthInMbs – cropping_rect_right] + pan_scan_rect_right_offset – 1 and with vertical coordinates from 16 * cropping_rect_top + pan_scan_rect_top_offset to 16 * [16 * PicHeightInMb s – cropping_rect_bottom] + pan_scan_rect_bottom_offset – 1, inclusive. If this rectangular area includes samples outside of the cropping rectangle, the region outside of the cropping rectangle may be filled with synthesized content (such as black video c ontent or neutral grey video content) for display.

The value of 16 * cropping_rect_left + pan_scan_rect_left_offset shall be less than or equal to 16 * [16 * PicWidthInMbs – cropping_rect_right] + pan_scan_rect_right_offset – 1.

The value of 16 * cropping_ rect_top + pan_scan_rect_top_offset shall be less than or equal to 16 * [16 * PicHeightInMbs – cropping_rect_bottom] + pan_scan_rect_bottom_offset – 1.

D.2.4 Filler payload SEI message semantics

This message contains a series of PayloadSize bytes of value 0xFF, which can be discarded.

filler_byte shall be a byte having the value 0xFF.

D.2.5 User data registered by ITU-T Recommendation T.35 SEI message semantics

This message contains registered user data as specified by ITU -T Recommendation T.35.

itu_t_t35_country_code shall be a byte having a value specified as a country code by ITU -T Recommendation T.35.

itu_t_t35_country_code_extension_byte shall be a byte having a value specified as an extended country code by ITU -T Recommendation T.35.

itu_t_t35_payload_byte shall be a byte containing user data registered as specified by ITU -T Recommendation T.35.

D.2.6 User data unregistered SEI message semantics

This message contains unregistered arbitrary user data, the contents of which are not specified by this Recommendation | International Standard.

user_data_arbitrary_payload_byte is a byte having a value not specified by this Recommendation | International Standard.

NOTE - Users of this Recommendation | International Standard should exercise care in the use of the user data arbitrary SEI message to avoid the carriage of data content in a form likely to conflict with the data content format of other users (e.g., avoiding conflict by using a fixed multi-byte prefix identifier within the payload content).

D.2.7 Random access point SEI message semantics

The random access point SEI message assists a decoder to start decoding at a picture other than the first picture in the bitstream. All decoded pictures at or subsequent to the recovery point in output order are indicated to be correct or approximately correct in content, if decoding is started at the first picture in decoding order after the random access point SEI message. Decoded pictures produced by starting the decoding process at the random access point may not be correct in content until the indicated recovery point, and the decoding process starting at the random access point and ending at the recovery point may contain references to pictures not available in the decoded picture buffer.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

In addition, by use of the broken_link_flag, th e random access point SEI message can indicate to the decoder the location of a splice point in the bitstream that can result in serious visual artifacts even when the decoding process was begun at a previous IDR point in decoding order.

The recovery point is indicated as a count in units of coded pictures subsequent to the current picture at the position of the SEI message.

recovery_frame_cnt indicates the recovery point of output pictures in output order. All decoded pictures in output order are indicated to be correct or approximately correct in content starting at the output order position of the reference picture having the frame number equal to the frame number of the next slice incremented by recovery_frame_cnt in modulo MaxFrameNum arithmetic. recove ry_framepoint_count shall not exceed MaxFrameNum -1.

exact_match_flag indicates whether decoded pictures at and subsequent to the recovery point in output order obtained by starting the decoding process at the specified random access point shall be an exact match to the pictures that would be produced by a decoder starting at the last prior IDR point in the NAL unit stream. The value 0 indicates that the match may not be exact and the value 1 indicates that the match shall be exact.

If decoding starts from an random access point indicated in a random access point SEI message, all references to unavailable reference pictures shall be inferred as references to sample values given by Y=Cb=Cr=128 (mid -level grey) in intra macroblocks for purposes of determining the conformance of the value of exact_match_flag.

broken_link_flag indicates the presence or absence of a splicing point in the NAL unit stream at the location of the random access point SEI message. If broken_link_flag is equal to 1, pictures produced by starting the decoding process at the last previous IDR point may contain undesirable visual artifacts due to splicing operations and should not be displayed until the indicated random access recovery point in output order. If broken_link_flag is equal to 0, no indication is given regarding any potential presence of visual artifacts.

Regardless of the value of the broken_link_flag, pictures subsequent to the recovery point in output order are indicated to be correct or approximately correct in content.

If a sub-sequence information SEI message is transmitted in conjunction with a random access point SEI message in which broken_link_flag is equal to 1 and if subseq_layer_num is 0, subseq_id should be different from the latest subseq_id for subseq_layer_num e qual to 0 that was decoded prior to the random access point. If broken_link_flag is equal to 0, the subseq_id in sub -sequence layer 0 should remain unchanged.

If HRD information is present in the bitstream, a buffering period SEI message shall be transmit ted after the random access point SEI message and before any slice data in order to establish initialisation of the HRD buffer model.

D.2.8 Decoded picture buffer management Repetition SEI message semantics

The Decoded picture buffer management repetition SEI me ssage is used to repeat memory management control operation commands that were located earlier in decoding order.

original_frame_num identifies the picture were the repeated memory management control operation originally occurred.

dec_pic_buffer_managamen t( ) shall contain a copy of the decoded picture buffer management syntax elements of the picture whose frame_num was original_frame_num.

D.2.9 Spare picture SEI message semantics

The spare picture SEI message indicates that certain macroblocks, called spare dec oded macroblocks, in one or more decoded reference pictures resemble the co -located macroblocks in a certain decoded picture, called the target picture, so much that any of these spare decoded macroblocks can be used to replace a co -located incorrect decod ed macroblock in the target picture in the multi -frame buffer and in decoder output. Decoded pictures that contain spare macroblocks are called spare pictures.

target_frame_num identifies the target picture whose spare pictures and macroblocks are specifi ed later in the message. The target picture is the latest reference picture having frame_num equal to target_frame_num.

num_spare_pics_minus1 specifies the number of pictures which contain spare picture or macroblocks for the target picture. num_spare_pic s_minus1 shall not exceed 14.

delta_spare_frame_num specifies to which spare picture the following spare picture information in the current loop count belongs. For the first spare picture of the message, CandidateSpareFrameNum is set to target_frame_num -1, if target_frame_num is greater than 0 and MaxFrameNum – 1 otherwise. For later spare pictures, CandidateSpareFrameNum is the SpareFrameNum of the previous loop round minus 1 if SpareFrameNum is greater than 0 and MaxFrameNum – 1 otherwise. For each loop r ound, SpareFrameNum is calculated as follows:

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

```
SpareFrameNum = CandidateSpareFrameNum – delta_spare_frame_num
if( SpareFrameNum < 0 )
        SpareFrameNum = MaxFrameNum + SpareFrameNum                    (D-3)
```

ref_area_indicator specifies how the locations of spare macroblocks are coded. ref_area_indicator 0 indicates that all macroblocks of the spare picture are spare macroblocks.. ref_area_indicator 1 indicates an uncompressed spare macroblock map. ref_area_indicator 2 indicates a compressed spare macr oblock map. A spare macroblock map consists of flags for each macroblock location of a picture. ref_area_indicator shall not exceed 2.

If ref_area_indicator is 1, there is a **ref_mb_indicator** for each macroblock address of the spare macroblock map in raster scan order. ref_mb_indicator 0 indicates that the macroblock is a spare macroblock, and ref_mb_indicator 1 indicates that the macroblock is not a spare macroblock.

If ref_area_indicator is 2, a spare macroblock map between a spare picture and the target picture is compressed. The coded macroblock map for loop_count equal to 0 is the spare macroblock map between the target picture and the first spare picture. A coded macroblock map for loop_count greater than 0 is generated by applying an exclusive or oper ation between the previous spare macroblock map and the current spare macroblock map. The coded macroblock map is scanned in counter-clockwise box-out order as specified in subclause 8.2.4.1. The number of consecutive zeros in t he scanning order is indicated in **zero_run_length**. [Ed. Note: This paragraphs needs work. What is loop_count? What does compressed mean? What does it mean for something to be between a spare picture and the target picture?]

D.2.10 Scene information SEI message semantics

A scene is herein defined as a set of consecutive pictures in output order captured with one camera. Decoded pictures within one scene generally have similar pictorial contents and semantic meanings.

The scene information SEI message is used to label scenes with identifiers and to indicate scene changes. The message indicates how the uncoded source picture for the coded target picture was created. The decoder may use the information to select an appropriate algorithm to conceal transmission erro rs. For example, a specific algorithm may be used to conceal transmission errors that occurred in pictures belonging to a gradual scene transition. Furthermore, the scene information SEI message may be used for external purposes, such as for indexing the s cenes of a coded sequence.

scene_id identifies the scene which the target picture belongs to. Pictures in a scene shall share the same value of scene_id. Consecutive scenes in output order should not have the same value of scene_id. If the next slice or s lice data partition in decoding order belongs to a picture that includes contents from two scenes, scene_id is the scene identifier of the former scene in output order.

scene_transition_type indicates in which type of a scene transition, if any, the target picture is involved. The valid values of **scene_transition_type** are specified in Table D-4. scene_transition_type shall not exceed 6. If a value greater than 6 is received by the decoder, it shall be ignored (removed from the bit stream and discarded).

Table D-4 – Scene transition types.

Value	Description
0	No transition
1	Fade to black
2	Fade from black
3	Unspecified transition from or to constant color
4	Dissolve
5	Wipe
6	Unspecified mixture of two scenes
Other values	Reserved

If scene_transition_type is greater than 3, the next slice or slice data partition in decoding order belongs to a picture that includes contents from two scenes.

Scene transition types are specified as follows. The term "the former scene" is used to indicate the former scene involved in a scene transition in output order. The term "the latter scene" is used to indicate the latter scene involved in a scene transition in output order.

"**No transition**" indicates that the target picture does not belong to a gradual scene transition (all scene transiton types but scene cut). If two consecutive pictures in output order have scene_transition_type equal to zero and different scene IDs, a scene cut occurred between the two pictures.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

In "**Fade to black**" scene transition, the lightness of the scene monotonically reduces to zero (i.e., black picture).

In "**Fade from black**" scene transition, the lightness of the scene monotonically increases from zero to full intensity.

An uncompressed picture in "**Dissolve**" scene transition is generated by calculating a sum of co -located weighted sample values of a picture from the former scene and a picture from the latter scene. The weight of the former scene gradually decreases from full level to zero level, whereas the weight of the latter scene gradually increases from zero level to full level.

In "**Wipe**" scene transition, the latter scene covers the former scene in a spatially gradual manner, while the intensity of the scenes remains unchanged.

second_scene_id is present if the next slice or slice data partition in decoding order belongs to a picture that includes contents from two scenes, i.e., scene_transition_type is larger than 3. second_scene_id is the scene identifier of the latter scene in output order.

D.2.11 Sub-sequence information SEI message semantics

The sub-sequence information SEI message is used to indicate the position of a picture in data dependency hierarchy that consists of sub-sequence layers and sub-sequences.

A sub-sequence layer contains a subset of the coded pictures in a sequence. Sub -sequence layers are numbered with non - negative integers. A layer having a larger layer number is a higher layer than a layer having a smaller layer number. The layers are ordered hierarchically based on their dependency on each other so that a layer does not depend on any higher layer and may depend on lower layers. In other words, layer 0 is independently decodable, pictures in layer 1 may be predicted from layer 0, pictures in layer 2 may be predicted from la yers 0 and 1, etc.

NOTE: The subjective quality is expected to increase along with the number of decoded layers.

A sub-sequence is a set of coded pictures within a sub -sequence layer. A picture shall reside in one sub -sequence layer and in one sub-sequence only. A sub-sequence shall not depend on any other sub -sequence in the same or in a higher sub-sequence layer. A sub-sequence in layer 0 can be decoded independently of any other sub -sequences and previous long-term reference pictures.

The sub-sequence information SEI message concerns the picture containing the next slice or slice data partition in decoding order. The picture which the next slice or slice data partition belongs to is herein referred to as the current picture.

When a sub-sequence information SEI message is coded, the required_frame_num_update_behaviour_flag in the active sequence parameter set shall be equal to 1.

subseq_layer_num indicates the sub-sequence layer number of the current picture. If subseq_layer_num is greater than 0, memory management control operations shall not be used in any slice header of the current picture. subseq_layer_num shall be less than 256.

subseq_id identifies the sub-sequence within a layer. Consecutive sub -sequences within a particular layer in decoding order shall have a different subseq_id from each other. subseq_id shall be less than 65536.

first_ref_pic_flag equal to 1 signals that the current picture is the first reference picture of the sub -sequence in decoding order. Otherwise the first_ ref_pic_flag sh all be 0.

leading_non_ref_pic_flag equals to 1 if the current picture is a non -reference picture preceding any reference picture in decoding order within the sub -sequence. Otherwise the leading_non_ref_pic_flag shall be 0.

last_pic_flag equal to 1 signals that the current picture is the last picture of the sub -sequence (in decoding order), including all reference and non -reference pictures of the sub-sequence. For any other pictures, the last_pic_flag shall be 0.

subseq_frame_num shall be equal to 0 for the first reference picture of the sub -sequence and for any non -reference picture preceding the reference picture of the sub -sequence in decoding order. For each coded picture belonging to the sub-sequence in decoding order, subseq_frame_num shall be incremen ted by 1, in modulo MaxFrameNum operation, relative to the previous reference frame that belongs to the sub -sequence. Both fields of a frame, if present, shall have the same subseq_frame_num. subseq_frame_num shall be less than MaxFrameNum.

If the current picture is an IDR picture, it shall start a new sub -sequence in sub-sequence layer 0. Thus, the subseq_layer_num shall be 0, the subseq_id shall be different from the previous sub -sequence in sub-sequence layer 0, first_ref_pic_flag shall be 1, and leadin g_non_ref_pic_flag shall be 0.

D.2.12 Sub-sequence layer characteristics SEI message semantics

The sub-sequence layer characteristics SEI message indicates the characteristics of sub -sequence layers.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

A pair of average bit rate and average frame rate characterizes each sub-sequence layer. The first pair of average bit rate and average frame rate signals the characteristics of sub-sequence layer 0. The second pair, if present, signals the characteristics of sub-sequence layers 0 and 1 jointly. Each pair in decoding order signals the characteristics for a range of sub-sequence layers from layer number 0 to the layer number that is incremented by one from the previous upper limit of layer numbers. The values are in effect from the point they are decoded until an update of the values is decoded.

accurate_statistics_flag indicates how reliable the values of average_bit_rate and average_frame_rate are. accurate_statistics_flag equal to 1 indicates that the average_bit_rate and the average_frame_rate are rounded from statistically correct values. accurate_statistics_flag equal to 0 indicates that the average_bit_rate and the average_frame_rate are estimates and may deviate somewhat from the correct values.

average_bit_rate gives the average bit rate in units of 1000 bits per second. All NAL units in the range of sub-sequence layers specified above are taken into account in the calculation. The average bit rate is calculated according to the picture removal time specified in Annex C of the Recommendation | International Standard. In the following, B is the number of bits in all NAL units succeeding a sub-sequence layer characteristics SEI message (including the bits of the NAL units of the current picture) and preceding the next sub-sequence layer characteristics SEI message or the end of the stream. t_1 is the removal time (in seconds) of the current picture, and t_2 is the removal time (in seconds) of the latest picture before the next sub-sequence layer characteristics SEI message or the end of the stream. Then, the average_bit_rate is calculated as follows provided that $t_1 \neq t_2$:

$$\text{average\_bit\_rate} = \text{round}(B \div ((t_2 - t_1) * 1000)) \quad \text{(D-4)}$$

If t1 = t2, average_bit_rate shall be 0. Value zero indicates an unspecified bit rate.

average_frame_rate gives the average frame rate in units of frames/(256 seconds). All NAL units in the range of sub-sequence layers specified above are taken into account in the calculation. In the following, C is the number of frames between the current picture (inclusive) and the next sub-sequence layer characteristics SEI message or the end of the stream. t_1 is the removal time (in seconds) of the current picture, and t_2 is the removal time (in seconds) of the latest picture before the next sub-sequence layer characteristics SEI message or the end of the stream. Then, the average_frame_rate is calculated as follows provided that $t_1 \neq t_2$:

$$\text{average\_frame\_rate} = \text{round}(C * 256 \div (t_2 - t_1)) \quad \text{(D-5)}$$

If $t_1 = t_2$, average_frame_rate shall be 0. Value zero indicates an unspecified frame rate.

D.2.13 Sub-sequence characteristics SEI message semantics

The sub-sequence characteristics SEI message indicates the characteristics of a sub-sequence. It also indicates inter prediction dependencies between sub-sequences.

subseq_layer_num indicates the sub-sequence layer number to which the sub-sequence characteristics SEI message applies. subseq_layer_num shall be less than 256.

subseq_id indicates the sub-sequence within a layer to which the sub-sequence characteristics SEI message applies. subseq_id shall be less than 65536.

This message applies to the next sub-sequence in decoding order having the indicated subseq_layer_num and subseq_id. This sub-sequence is herein called the target sub-sequence.

duration_flag equal to 0 indicates that the duration of the target sub-sequence is not specified.

subseq_duration indicates the duration of the target sub-sequence in clock ticks of a 90-kHz clock.

average_rate_flag equal to 0 indicates that the average bit rate and the average frame rate of the target sub-sequence are unspecified.

accurate_statistics_flag indicates how reliable the values of average_bit_rate and average_frame_rate are. accurate_statistics_flag equal to 1, indicates that the average_bit_rate and the average_frame_rate are rounded from statistically correct values. accurate_statistics_flag equal to 0 indicates that the average_bit_rate and the average_frame_rate are estimates and may deviate from the statistically correct values.

average_bit_rate gives the average bit rate in (1000 bits)/second of the target sub-sequence. All NAL units of the target sub-sequence are taken into account in the calculation. The average bit rate is calculated according to the picture removal time specified in subclause C.1.1.2. In the following, B is the number of bits in all NAL units in the sub-sequence. t_1 is the removal time (in seconds) of the first picture of the sub-sequence (in decoding order), and t_2 is the removal time (in seconds) of the last picture of the sub-sequence (in decoding order). Then, the average_bit_rate is calculated as follows provided that $t_1 \neq t_2$:

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

$$average\_bit\_rate = round(B \div ((t_2 - t_1) * 1000)) \quad (D\text{-}6)$$

If $t_1 = t_2$, average_bit_rate shall be 0.

average_frame_rate gives the average frame rate in units of frames/(256 seconds) of the target sub-sequence. All NAL units of the target sub-sequence are taken into account in the calculation. The average frame rate is calculated according to the picture removal time specified in subclause C.1.1.2. In the following, C is the number of frames in the sub-sequence. t_1 is the removal time (in seconds) of the first picture of the sub-sequence (in decoding order), and t_2 is the removal time (in seconds) of the last picture of the sub-sequence (in decoding order). Then, the average_frame_rate is calculated as follows provided that $t_1 \neq t_2$:

$$average\_frame\_rate = round(C * 256 \div (t_2 - t_1)) \quad (D\text{-}7)$$

If t1 = t2, average_frame_rate shall be 0.

num_referenced_subseqs gives the number of sub-sequences which contain pictures that are used as reference pictures for inter prediction in the pictures of the target sub-sequence. num_referenced_subseqs shall be less than 256.

ref_subseq_layer_num and **ref_subseq_id** identify a sub-sequence that contains pictures that are used as reference pictures for inter prediction in the pictures of the target sub-sequence.

D.2.14 Full-frame freeze SEI message semantics

The full-frame freeze SEI message indicates that the contents of the entire prior displayed video frame in output order should be kept unchanged, without updating the display using the contents of the current decoded picture. The displayed frame should then remain unchanged until a full-frame freeze release SEI message is received, or until timeout occurs, whichever comes first. The full-frame freeze shall lapse due to timeout after five seconds or five pictures in output order, whichever is a longer period of time. The timeout can be prevented by the issuance of another full-frame freeze SEI message prior to or upon expiration of the timeout period.

D.2.15 Full-frame freeze release SEI message semantics

The full-frame freeze release SEI message indicates that the update of the displayed video frame should resume, starting with the contents of the current decoded picture and continuing for subsequent pictures in output order. The full-frame freeze release SEI message cancels the effect of any full-frame freeze SEI message sent with pictures that precede the current picture in output order.

D.2.16 Full-frame snapshot SEI message semantics

The full-frame snapshot SEI message indicates that the current frame is labeled for external use as a still-image snapshot of the video content.

snapshot_id specifies a snapshot identification number. snapshot_id shall not exceed 0xFFFFFFFF .

Values of snapshot_id from 0 to 0x7FFFFFFF may be used as specified externally. Values of snapshot_id from 0x80000000 to 0xFFFFFFFF are reserved for future use by ITU-T | ISO/IEC. Decoders encountering a value of snapshot_id from 0x80000000 to 0xFFFFFFF F shall ignore (remove from the bitstream and discard) it.

D.2.17 Progressive refinement segment start SEI message semantics

The progressive refinement segment start SEI message indicates the beginning of a specified set of consecutive coded pictures which is labeled as the current picture followed by a sequence of one or more pictures of refinement of the quality of the current picture, rather than as a representation of a continually moving scene.

The tagged set of consecutive coded pictures shall continue until one of following conditions is true. When a condition below becomes true, the next slice to be decoded does not belong to the tagged set of consecutive coded pictures.

1. The next slice to be decoded belongs to an IDR picture.
2. num_refinement_steps_minus1 is greater than 0 and the frame number of the next slice to be decoded is (CurrFrameNum + num_refinement_steps_minus1 + 1) % MaxFrameNum, where CurrFrameNum is the frame number of the next slice following this SEI message in decoding order.
3. num_refinement_steps_minus1 is 0 and a progressive refinement segment end SEI message with the same progressive_refinement_id as the one in this SEI message is decoded.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

4. num_refinement_steps_minus1 is 0 and a timeout of five seconds has elapsed since the decoding of this S EI message. The timeout can be prevented by the issuance of an identical progressive refinement segment start SEI message prior to or upon expiration of the timeout period.

The decoding order of picture within the tagged set of consecutive pictures should be the same as their output order. **progressive_refinement_id** specifies an identification number for the progressive refinement operation. progressive_refinement_id shall not exceed 0xFFFFFFFF.

Values of progressive_refinement_id from 0 to 0x7FFFFFFF may b e used as specified externally. Values of progressive_refinement_id from 0x80000000 to 0xFFFFFFFF are reserved for future use by ITU -T | ISO/IEC. Decoders encountering a value of progressive_refinement_id from 0x80000000 to 0xFFFFFFFF shall ignore (remove from the bitstream and discard) it.

num_refinement_steps_minus1 indicates the number of reference frames in the tagged set of consecutive coded pictures. num_refinement_steps equal to 0 indicates that number of reference frames in the tagged set of consec utive coded pictures is unknown. Otherwise, the number of reference frames in the tagged set of consecutive coded pictures is equal to num_refinement_steps_minus1 + 1. num_refinement_steps_minus1 shall not exceed MaxFrameNum -1.

D.2.18 Progressive refinement segment end SEI message semantics

The progressive refinement segment end SEI message indicates the end of a specified set of consecutive coded pictures which has been labeled by use of a progressive refinement segment start SEI message as an initial picture fol lowed by a sequence of one or more pictures of the refinement of the quality of the initial picture, and ending with the current picture.

progressive_refinement_id specifies an identification number for the progressive refinement operation. progressive_refinement_id shall not exceed 0xFFFFFFFF.

The progressive refinement segment end SEI message indicates the end of any progressive refinement segment previously started using a progressive refinement segment start SEI message with the same value of progressive_refinement_id.

Values of progressive_refinement_id from 0 to 0x7FFFFFFF may be used as specified externally. Values of progressive_refinement_id from 0x80000000 to 0xFFFFFFFF are reserved for future use by ITU -T | ISO/IEC. Decoders encountering a value of progressive_refinement_id from 0x80000000 to 0xFFFFFFFF shall ignore (remove from the bitstream and discard) it.

D.2.19 Reserved SEI message semantics

This message consists of data reserved for future backward -compatible use by ITU-T | ISO/IEC. Encoders conforming to this Recommendation | International Standard shall not send reserved SEI messages until and unless the use of such messages has been specified by ITU-T | ISO/IEC. Decoders conforming to this Recommendation | International Standard that encounter reserved SEI messages shall discard their content without effect on the decoding process, except as specified in future Recommendations | International Standards specified by ITU -T | ISO/IEC. reserved_sei_message_payload_byte is a byte reserved for future use by ITU-T | ISO/IEC.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Annex E
Video usability information

(This annex forms an integral part of this Recommendation | International Standard)

This Annex specifies syntax and semantics of those parts of the sequence parameter set and the picture param eter set that are not required for determining the decoded values of samples. The parameters specified in this annex can be used to facilitate the use of the decoded pictures or facilitate the resource allocation of a decoder by restricting certain video parameters beyond those limits specified by Annex A. Decoders are not required to process VUI sequence parameters for conformance to this Recommendation | International Standard.

For some of the parameters of this Annex, default values are specified in the semantics subclause. The syntax includes flags that allow avoiding the signalling of groups of parameters. If a specific group of parameters is not coded, the default values for these parameters become effective.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

E.1 VUI syntax

E.1.1 VUI sequence parameters syntax

vui_seq_parameters( ) {	C	Descriptor
aspect_ratio_info_present_flag	0	u(1)
if(aspect_ratio_info_present_flag) {		
aspect_ratio_idc	0	u(8)
if(aspect_ratio_idc == Extended_SAR) {		
sar_width	0	u(16)
sar_height	0	u(16)
}		
}		
overscan_info_present_flag	0	u(1)
if(overscan_info_present_flag)		
overscan_appropriate_flag	0	u(1)
video_signal_type_present_flag	0	u(1)
if(video_signal_type_present_flag) {		
video_format	0	u(3)
video_full_range_flag	0	u(1)
colour_description_present_flag	0	u(1)
if(colour_description_present_flag) {		
colour_primaries	0	u(8)
transfer_characteristics	0	u(8)
matrix_coefficients	0	u(8)
}		
}		
chroma_location_info_present_flag	0	u(1)
if (chroma_location_info_present_flag) {		
chroma_location_frame	0	ue(v)
chroma_location_field	0	ue(v)
}		
timing_info_present_flag	0	u(1)
if(timing_info_present_flag) {		
num_units_in_tick	0	u(32)
time_scale	0	u(32)
fixed_frame_rate_flag	0	u(1)
}		
nal_hrd_parameters_present_flag	0	u(1)
if(nal_hrd_parameters_present_flag == 1)		
hrd_parameters( )		
vcl_hrd_parameters_present_flag	0	u(1)
if(vcl_hrd_parameters_present_flag == 1)		
hrd_parameters( )		
if((nal_hrd_parameters_present_flag == 1 || (vcl_hrd_parameters_present_flag == 1))		
low_delay_hrd_flag	0	u(1)
bitstream_restriction_flag	0	u(1)
if(bitstream_restriction_flag) {	0	u(1)

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

motion_vectors_over_pic_boundaries_flag	0	u(1)
max_bytes_per_pic_denom	0	ue(v)
max_bits_per_mb_denom	0	ue(v)
log2_max_mv_length_vertical	0	ue(v)
log2_max_mv_length_horizontal	0	ue(v)
max_dec_frame_reordering	0	ue(v)
max_dec_frame_buffering	0	ue(v)
}		
}		

E.1.2 HRD parameters syntax

hrd_parameters( ) { /* coded picture buffer parameters */	C	Descriptor
cpb_cnt	0	ue(v)
bit_rate_scale	0	u(4)
cpb_size_scale	0	u(4)
for(k=1; k<=cpb_cnt; k++) {		
bit_rate_value[k]	0	ue(v)
cpb_size_value[k]	0	ue(v)
vbr_cbr_flag[k]	0	u(1)
}		
initial_cpb_removal_delay_length_minus1	0	u(5)
cpb_removal_delay_length_minus1	0	u(5)
dpb_output_delay_length_minus1	0	u(5)
time_offset_length	0	u(5)
}		

E.1.3 VUI picture parameters syntax

vui_pic_parameters( ) {	C	Descriptor
frame_cropping_flag	1	u(1)
if(frame_cropping_flag) {		
frame_cropping_rect_left_offset	1	ue(v)
frame_cropping_rect_right_offset	1	ue(v)
frame_cropping_rect_top_offset	1	ue(v)
frame_cropping_rect_bottom_offset	1	ue(v)
}		
}		

E.2 VUI semantics

E.2.1 VUI sequence parameters semantics

aspect_ratio_info_present_flag : A flag that, when equal to 1, signals the presence of the aspect_ratio_idc. If the flag is 0, then the default value 0 shall apply to aspect_ratio_idc.

aspect_ratio_idc specifies the value of the sample aspect ratio. Table E-1 shows the meaning of the code. If aspect_ratio_idc indicates Extended_SAR, the sample aspect ratio is represented by sar_width and sar_height.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table E-1 – Meaning of sample aspect ratio indicator

aspect_ratio_idc	Sample aspect ratio	(informative) Examples of use
0	Unspecified or specified externally	
1	1:1 ("square")	1280x720 16:9 frame without overscan 1920x1080 16:9 frame without overscan (cropped from 1920x1088) 640x480 4:3 frame without overs can
2	12:11	720x576 4:3 frame with overscan 352x288 4:3 frame without overscan
3	10:11	720x480 4:3 frame with overscan 352x240 4:3 frame without overscan
4	16:11	720x576 16:9 frame with overscan 540x576 4:3 frame with overscan
5	40:33	720x480 16:9 fram e with overscan 540x480 4:3 frame with overscan
6	24:11	352x576 4:3 frame without overscan 540x576 16:9 frame with overscan
7	20:11	352x480 4:3 frame without overscan 480x480 16:9 frame with overscan
8	32:11	352x576 16:9 frame without overscan
9	80:33	352x480 16:9 frame without overscan
10	18:11	480x576 4:3 frame with overscan
11	15:11	480x480 4:3 frame with overscan
12	64:33	540x576 16:9 frame with overscan
13	160:99	540x480 16:9 frame with overscan
14-254	Reserved	
255	Extended_SAR	

sar_width indicates the horizontal size of the sample aspect ratio. A zero value is forbidden.

sar_height indicates the vertical size of the sample aspect ratio. A zero value is forbidden.

The sar_width and sar_height shall be relatively prime. If aspect_ratio_idc is zero or if either of sar_width or sar_height are zero, the sample aspect ratio shall be considered unspecified or specified externally.

overscan_info_present_flag: A flag that, when equal to 1, signals the presence of overscan_appropriate_flag. If overscan_info_present_flag is equal to 0 or is not present, the preferred display method for the video signal is unspecified.

overscan_appropriate_flag: A flag that, when equal to 1, indicates that the video signal is suitable for display using overscan. When overscan_appropriate_flag is equal to 0, this indicates that the video signal contains visually important information in the entire region out to the edges of the cropping rectangle of the picture, such that the video signal should not be displayed using overscan; instead it should be displayed using either an exact match between the display area and the cropping rectangle, or using underscan. For example, overscan_appropriate_flag equal to 1 might be used for entertainment television programming, or for a live view of people in a videoconference, and overscan_appropriate_flag equal to 0 might be used for computer screen capture or security camera content.

video_signal_type_present_flag: A flag that, when 1, signals the presence of video signal informatio n. If video_signal_type_present_flag is 0, then the following default values shall apply: video_format = 5, video_full_range_flag = 0, colour_description_present_flag = 0.

video_format: Indicates the representation of the pictures before being coded in accordance with this Recommendation | International Standard. Its meaning is specified in Table E-2.

Table E-2 – Meaning of video_format

video_format	Meaning
0	Component

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

1	PAL
2	NTSC
3	SECAM
4	MAC
5	Unspecified video format
6	Reserved
7	Reserved

video_full_range_flag indicates the nominal black level and range of the luma and chroma signals as derived from E'_Y, E'_{PB}, and E'_{PR} analogue component signals as follows:

If video_full_range_flag is equal to 0:

$$Y = \text{round}(219 * E'_Y + 16) \quad \text{(E-1)}$$

$$Cb = \text{round}(224 * E'_{PB} + 128) \quad \text{(E-2)}$$

$$Cr = \text{round}(224 * E'_{PR} + 128) \quad \text{(E-3)}$$

If video_full_range_flag is equal to 1:

$$Y = \text{round}(255 * E'_Y) \quad \text{(E-4)}$$

$$Cb = \text{round}(255 * E'_{PB} + 128) \quad \text{(E-5)}$$

$$Cr = \text{round}(255 * E'_{PR} + 128) \quad \text{(E-6)}$$

colour_description_present_flag indicates the presence of colour_primaries, transfer_characteristics and matrix_coefficients in the bitstream.

colour_primaries: This 8-bit integer describes the chromaticity coordinates of the source primaries, and is specified in Table E-3.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

Table E-3 – Colour Primaries

Value	Primaries		
0	Reserved		
1	ITU-R Recommendation BT.709		
	primary	x	y
	green	0.300	0.600
	blue	0.150	0.060
	red	0.640	0.330
	white D65	0.3127	0.3290
2	Unspecified or externally-specified video Image characteristics are unknown or specified externally.		
3	Reserved		
4	ITU-R Recommendation BT.470-2 System M		
	primary	x	y
	green	0.21	0.71
	blue	0.14	0.08
	red	0.67	0.33
	white C	0,310	0.316
5	ITU-R Recommendation BT.470-2 System B, G		
	primary	x	y
	green	0.29	0.60
	blue	0.15	0.06
	red	0.64	0.33
	white D65	0.3127	0.3290
6	Society of Motion Picture and Television Engineers 170M		
	primary	x	y
	green	0.310	0.595
	blue	0.155	0.070
	red	0.630	0.340
	white D65	0.3127	0.3290
7	Society of Motion Picture and Television Engineers 240M (1987)		
	primary	x	y
	green	0.310	0.595
	blue	0.155	0.070
	red	0.630	0.340
	white D65	0.3127	0.3290
8	Generic film (colour filters using Illuminant C)		
	primary	x	y
	green	0.243	0.692 (Wratten 58)
	blue	0.145	0.049 (Wratten 47)
	red	0.681	0.319 (Wratten 25)
9-255	Reserved		

If video_signal_type_present_flag is 0 or colour_description_present_flag is 0, colour_primaries shall have an inferred value equal to 2 (the chromaticity is unspecified or specified externally).

transfer_characteristics: This 8-bit integer describes the opto-electronic transfer characteristic of the source picture, and is specified in Table E-4.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

Table E-4 – Transfer Characteristics

Value	Transfer Characteristic
0	Reserved
1	ITU-R Recommendation BT.709 $V = 1.099\ L_C^{0.45} - 0.099$ for $1 \geq L_C \geq 0.018$ $V = 4.500\ L_C$ for $0.018 > L_C \geq 0$
2	Unspecified or externally-specified video Image characteristics are unknown or specified externally.
3	Reserved
4	ITU-R Recommendation BT.470-2 System M Assumed display gamma 2.2
5	ITU-R Recommendation BT.470-2 System B, G Assumed display gamma 2.8
6	Society of Motion Picture and Television Engineers 170M $V = 1.099\ L_C^{0.45} - 0.099$ for $1 \geq L_C \geq 0.018$ $V = 4.500\ L_C$ for $0.018 > L_C \geq 0$
7	Society of Motion Picture and Television Engineers 240M (1987) $V = 1.1115\ L_C^{0.45} - 0.1115$ for $L_C \geq 0.0228$ $V = 4.0\ L_C$ for $0.0228 > L_C$
8	Linear transfer characteristics i.e. $V = L_C$
9	Logarithmic transfer characteristic (100:1 range) $V = 1.0 - \log_{10}(L_C)/2$ for $1 = L_C = 0.01$ $V = 0.0$ for $0.01 > L_C$
10	Logarithmic transfer characteristic (316.22777:1 range) $V = 1.0 - \log_{10}(L_C)/2.5$ for $1 = L_C = 0.0031622777$ $V = 0.0$ for $0.0031622777 > L_C$
11-255	Reserved

If video_signal_type_present_flag is zero or colour_description_present_flag is 0, transfer_characteris tics shall have an inferred value equal to 2 (the transfer characteristics are unspecified or are specified externally).

matrix_coefficients: This 8-bit integer describes the matrix coefficients used in deriving luma and chroma signals from the green, blue, and red primaries, as specified in Table E-5.

Using the following definitions:

E'_R, E'_G, and E'_B are analogue with values between 0 and 1.

White is specified as $E'_R = E'_G = E'_B = 1$.

Then:

$$E'_Y = K_R * E'_R + (1 - K_R - K_B) * E'_G + K_B * E'_B \qquad \text{(E-7)}$$

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

$$E'_{PB} = 0.5 * (E'_R - E'_Y) \div (1-K_R) \quad \text{(E-8)}$$

$$E'_{PB} = 0.5 * (E'_B - E'_Y) \div (1-K_B) \quad \text{(E-9)}$$

Then E'_Y is analogue with values between 0 and 1, E'_{PB} and E'_{PR} are analogue with values between -0.5 and 0.5, and white is equivalently given by $E'_Y = 1$, $E'_{PB} = 0$, $E'_{PR} = 0$.

Table E-5 – Matrix Coefficients

Value	Matrix
0	Reserved
1	ITU-R Recommendation BT.709 $K_R = 0.2126$; $K_B = 0.0722$
2	Unspecified or externally-specified video Image characteristics are unknown or specified externally.
3	Reserved
4	Federal Communications Commission $K_R = 0.30$; $K_B = 0.11$
5	ITU-R Recommendation BT.470-2 System B, G: $K_R = 0.299$; $K_B = 0.114$
6	Society of Motion Picture and Television Engineers 170M $K_R = 0.299$; $K_B = 0.114$
7	Society of Motion Picture and Television Engineers 240M (1987) $K_R = 0.212$; $K_B = 0.087$
8-255	Reserved

If video_signal_type_present_flag is zero or colour_description_present_flag is 0, matrix_coeffi cients shall have an inferred value equal to 2 (the matrix coefficients shall be inferred to be unspecified or specified externally).

chroma_location_info_present_flag: A flag that, when 1, signals the presence of the chroma location information. If the flag is 0, then the following default values shall apply: chroma_location_frame = 0, chroma_location_field = 0; and the chroma locations in a frame shall be inferred to be the nominal locations shown in in Figure 6-1 and the chroma locations in a field shall be inferred to be the nominal locations shown in Figure 6-2.

chroma_location_frame specifies the 4:2:0 sampling structure for frame pictures according to Table E-6 and Figure E-1. The value of chroma_location_frame shall not exceed 3.

Table E-6 – Chroma Sampling Structure for Frame Pictures

Value of chroma_location_frame	Sampling Structure
0	unspecified
1	Frame according to Figure E-1 Chroma Sample Mode 1
2	Frame according to Figure E-1 Chroma Sample Mode 2
3	Frame according to Figure E-1 Chroma Sample Mode 3

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

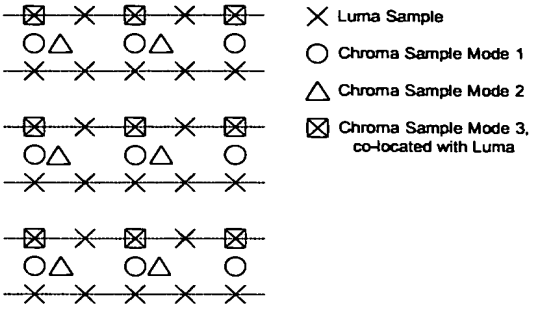


Figure E-1 – Luma and chroma sample types for frames

chroma_location_field specifies the 4:2:0 sampling structure for fields according to Table E-7 and Figure E-2. The value of chroma_location_field sh all not exceed 3.

Table E-7 – Chroma Sampling Structure for Fields

chroma_location_field	Sampling Structure
0	unspecified
1	Field according to Figure E-2 Chroma Sample Mode 1
2	Field according to Figure E-2 Chroma Sample Mode 2
3	Field according to Figure E-2 Chroma Sample Mode 3

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

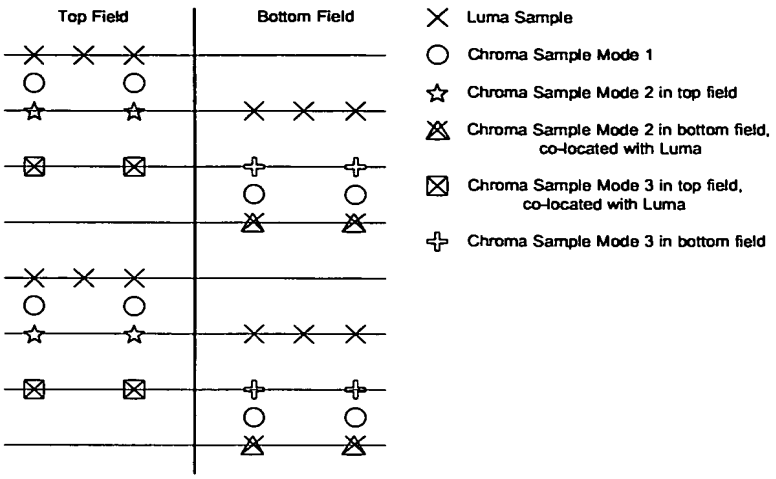


Figure E-2 – Luma and chroma association for fields

timing_info_ present_flag: A flag that, when equal to 1, signals the presence of time unit information. If timing_info_present_flag is equal to 0, then the following default values shall apply: num_units_in_tick = 0, time_scale = 0, fixed_frame_rate_flag = 0. [Ed. Note: These values make no sense, as it is not possible to create an HRD-conforming bitstream using these values.]

num_units_in_tick is the number of time units of a clock operating at the frequency time_scale Hz that corresponds to one increment of a clock tick counter. A clock tick is the minimum interval of time that can be represented in the coded data. For example, if the clock frequency of a video signal is (30 000) ÷ 1001 Hz, time_scale may be 30 000 and num_units_in_tick may be 1001. If num_units_in_tick is 0, the du ration of the clock tick is unspecified. [Ed. Note: This value makes no sense, as it is not possible to create an HRD -conforming bitstream without any specified clock tick.]

time_scale is the number of time units which pass in one second. For example, a ti me coordinate system that measures time using a 27 MHz clock has a time_scale of 27 000 000. If time_scale is 0, the duration of the clock tick specified above is unspecified. [Ed. Note: This value makes no sense, as it is not possible to create an HRD -conforming bitstream without any specified time scale.]

fixed_frame_rate_flag: A bit that, if equal to 1, indicates that the temporal distance between the HRD output times of any two consecutive frames or fields in output order as specified in Annex C is a co nstant. If equal to 0, the temporal distances between HRD output times of consecutive frames or fields in output order as specified in Annex C may not be constant.

nal_hrd_parameters_present_flag: If nal_hrd_parameters_present_flag is equal to 0, no HRD parameters pertaining to the multiplexed NAL and VCL bitstream are present in the bitstream. In this case, the multiplexed NAL and VCL bitstream conforms to an HRD whose buffer size and bit rate parameters are 20% higher than those given by the maxima specified in Table A-1 for the level specified by level_idc.

Note – when nal_hrd_parameters_present_flag is equal to 0, the conformance of the multiplexed NAL and VCL bitstream cannot be verified without external provision of all the NAL HRD parameters, including the NAL sequence HRD parameter information and all buffering period and picture timing SEI messages.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

If nal_hrd_parameters_present_flag is equal to 1, then NAL sequence HRD parameters (subclauses E.1.2 and E.2.2) immediately follow the flag. At least one of the NAL HRD coded picture buffers signalled in the sequence HRD parameters must have CPB bitrate and buffer size values that are both less than values 20% higher than the maxima specified in Table A-1 for the level specified by level_idc. Some of the NAL HRD coded picture buffers signalled may have values outside the level limits.

Furthermore, if nal_hrd_parameters_present_flag is equal to 1, then a buffering period SEI message (subclauses D.1.1 and D.2.1) immediately follows each random access point SEI message (subclauses D.1.7 and D.2.7), and a picture timing SEI message (subclauses D.1.1 and D.2.1) precedes each coded picture.

vcl_hrd_parameters_present_flag: If vcl_hrd_parameters_present_flag is equal to 0, no HRD parameters pertaining to the video coding layer are present in the bitstream. In this case, the bitstream conforms to an HRD whose buffer size and bit rate parameters are given by the maxima specified in Table A -1 for the level specified by level_idc.

Note – when vcl_hrd_parameters_present_flag is equal to 0, the conformance of the VCL bitstream cannot be verified without external provision of all the VCL HRD parameters, including the VCL sequence HRD parameter information and all buffering period and picture timing SEI messages.

If vcl_hrd_parameters_present_flag is equal to 1, then VCL sequence HRD parameters (subclauses E.1.2 and E.2.2) immediately follow the flag. At least one of the VCL HRD coded picture buffers signalled in the sequence HRD parameters must have CPB bitrate and buffer size values that are both less than the maxima specified in Table A -1 for the level specified by level_idc. Some of the HRD coded picture buffers signalled may have values outside the level limits.

Furthermore, if vcl_hrd_parameters_present_flag is equal to 1, then a buffering period SEI message (subclauses D.1.1 and D.2.1) immediately follows each random access point SEI message (subclauses D.1.7 and D.2.7), and a picture timing SEI message (subclauses D.1.1 and D.2.1) precedes each coded picture.

low_delay_hrd_flag: If low_delay_hrd_flag is equal to 0, the HRD operates in delay-tolerant mode. If low_delay_hrd_flag is equal to 1, the HRD operates in low-delay mode. In low-delay mode, only one HRD buffer may be selected and big pictures which violate the HRD removal time rules at the CPB are permitted.

NOTE - It is expected, but not required, that such big pictures occur only occasionally.

bitstream_restriction_flag: A flag that, when 1, signals the presence of bitstream restriction information. If bitstream_restriction_flag is set to 0, then the following default values shall apply: motion_vectors_over_pic_boundaries_flag = 1, max_bytes_per_pic_denom = 4, max_bits_per_mb_denom = 1, log2_max_mv_length_vertical = 16, log2_max_mv_length_horizontal = 16, max_dec_frame_buffering = 15, max_dec_frame_reordering = 15.

motion_vectors_over_pic_boundaries_flag equal to 0 indicates that no motion vector refers to samples outside the picture boundaries of the reference picture. motion_vectors_over_pic_boundaries_flag equal to 1 indicates that motion vectors may refer to samples outside the picture boundaries of a reference picture.

max_bytes_per_pic_denom indicates the maximum number of bytes that represent a picture.

The number of bytes that represent a picture in the NAL unit stream is defined for this purpose as the total number of bytes of NAL unit data (i.e., the total of the NumBytesInNALunit variables for the picture) that contain slices and slice data partitions for the picture. The value of max_bytes_per_pic_denom shall not exceed 16.

If max_bytes_per_pic_denom is equal to 0, no limit is indicated.

If max_bytes_per_pic_denom is not equal to 0, no coded frame shall be represented in the sequence by more than

(PicWidthInMbs*PicHeightInMbs)*(256*ChromaFormatParameter) ÷max_bytes_per_pic_denom

bytes.

If max_bytes_per_pic_denom is not equal to 0, no coded field shall be represented in the sequence by more than

(PicWidthInMbs*PicHeightInMbs)*(128*ChromaFormatParameter) ÷max_bytes_per_pic_denom

bytes.

[Ed. Note (Karsten): This references „ChromaFormatParameter" from table A -2. But there is no transmitted chroma format, so do we have to guess here? No – ChromaFormatParameter is always 4:2:0, as specified in Annex A]

max_bits_per_mb_denom indicates the maximum number of coded bits that represent a macroblock. The value of max_bits_per_mb_denom shall not exceed 16.

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

If max_bits_per_mb_denom is equal to 0, no limit is indicated. If max_bits_per_mb_denom is not equal to 0, no coded macroblock shall be represented in the bitstream by more than

(2048 * ChromaFormatParameter) ÷ max_bits_per_mb_denom

bits.

[Ed. Note: Insert reference to method of counting bits for a particular MB.]

[Ed. Note (Karsten): This references „ChromaFormatParameter" from table A-2. But there is no transmitted chroma format, so do we have to guess here? No – ChromaFormatParameter is always 4:2:0, as specified in Annex A]

log2_max_mv_length_vertical and **log2_max_mv_length_horizontal** indicate the maximum value of the absolute value of a decoded vertical or horizontal motion vector component, in ¼ luma sample unit s. A value of n asserts that no absolute value of a motion vector component is bigger than 2^n units of ¼ sample displacement. The value of log2_max_mv_length_vertical shall not exceed 16. The value of log2_max_mv_length_horizontal shall not exceed 16.

NOTE - The maximum absolute value of a decoded vertical or horizontal motion vector component is is also constrained by profile and level limits as specified in Annex A.

max_dec_frame_reordering specifies the maximum extent of reordering of frames from decod ing order to output order in the sequence. The sequence shall not require a decoded picture buffer with capacity of more than max_dec_frame_reordering frames to enable the output of decoded pictures in the output order indicated by the picture order count. The value of max_dec_frame_reordering shall not exceed the value of max_dec_frame_buffering.

For example, if max_dec_frame_reordering is equal to 0, then the output order of pictures in the bitstream shall be the same as the decoding order. If max_dec _frame_reordering is equal to 2, then no more than two pictures that follow any picture in decoding order may precede it in output order.

max_dec_frame_buffering specifies the maximum extent of buffering of frames from decoding time to output time in the sequence. The sequence shall not require a decoded picture buffer with capacity of more than max_dec_frame_buffering frames to enable the output of decoded pictures at the output times of the HRD. The value of max_dec_frame_reordering shall not exceed 15.

E.2.2 HRD parameters semantics

If multiple sequence parameter sets pertain to the bitstream, they shall contain consistent HRD information. [Ed. Note: What is the definition of consistent?]

cpb_cnt: This syntax element indicates the number of CPBs in the HRD. The value of cpb_cnt shall not be equal to 0 and shall not exceed 32. [Ed. Note: Should send cpb_cnt_minus1 rather than cpb_cnt.]

bit_rate_scale: Together with bit_rate_value[k], this syntax element specifies the maximum input bit rate of the k -th CPB in an HRD.

cpb_size_scale is used together with cpb_size_value[k] to define the CPB size of the k -th CPB in an HRD.

bit_rate_value[k]: Together with bit_rate_scale, this syntax element specifies the maximum input bit rate of the k -th CPB in an HRD. bit_rate_value[k] shall not exceed 2^{32}-1. The actual bit rate in bits per second is given by:

$$bit\_rate[\,k\,] = bit\_rate\_value[k] * 2^{(6 + bit\_rate\_scale)}. \quad \text{(E-1)}$$

cpb_size_value[k] is used together with cpb_size_scale to define the ma ximum input bit rate of the k-th CPB in an HRD. cpb_size_value[k] shall not exceed 2^{32}-1.

The actual buffer size in bits is given by

$$cpb\_size[k] = cpb\_size\_value[k] * 2^{(4 + cpb\_size\_scale)}. \quad \text{(E-2)}$$

For VCL HRD parameters, there shall be one value of k for which bit_rate[k] is within the maximum video bitrate, and cpb_size[k] is within the maximum CPB size specified in Table A-1. For NAL HRD parameters, there must be one value of k for which bit_rate[k] is within 120% of the maximum video bitrate, and cpb_size[k] is within 120% of the maximum CPB size.

vbr_cbr_flag: If equal to 0, this syntax element indicates that the CPB operates in variable bit rate (VBR) mode. If equal to 1, it indicates constant bit rate (CBR) operation.

Deleted: DRAFT ISO/IEC 14496-10 : 2002 (E)

initial_cpb_removal_delay_length_minus1 : Specifies the length in bits of the initial_cpb_removal_delay syntax element. The length of initial_cpb_removal_delay is initial_cpb_removal_delay_length_minus1+1..

cpb_removal_delay_length_minus1: Specifies the length in bits of the cpb_removal_delay syntax element. The length of cpb_removal_delay is cpb_removal_delay_length_minus1+1.

dpb_output_delay_length_minus1 : Specifies the length in bits of the dpb_output_delay syntax element. The lengt h of dpb_output_delay is dpb_output_delay_length_minus1+1.

time_offset_length: Specifies the length in bits of the time_offset syntax element.

E.2.3 VUI picture parameters semantics

frame_cropping_flag when equal to 1, signals the presence of frame cropping inf ormation. If frame_cropping_flag is set to 0, then the following default values shall apply frame_cropping_rect_left = 0, frame_cropping_rect_right = 0, frame_cropping_rect_top = 0, frame_cropping_rect_bottom = 0.

frame_cropping_rect_left, frame_cropping_ rect_right, frame_cropping_rect_top, frame_cropping_rect_bottom define the area of the luma frame internal array which shall contain the output region of the decoding process. The decoded values of these offsets consist of non -negative integer values, and the output region of the decoding process is specified as the area within the frame rectangle containing luma samples with horizontal coordinates from frame_cropping_rect_left to 16*PicWidthInMbs -(frame_cropping_rect_right + 1) and with vertical coordinates from frame_cropping_rect_top to 16*PicHeightInMbs -(frame_cropping_rect_bottom + 1), inclusive.

The value of frame_cropping_rect_left shall not exceed 16*PicWidthInMbs -(frame_cropping_rect_right – 1).

The value of frame_cropping_rect_top shall not exceed 16*PicHeightInMbs -(frame_cropping_rect_bottom – 1).

Deleted: DRAFT ITU-T Rec. H.264 (2002 E)

INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 1997 - 2000

COM 99-E
mmm yyyy
Original: English

Question: 15/16

STUDY GROUP 16 - CONTRIBUTION 999

SOURCE*: Gary Sullivan, Q.15/16 Rapporteur

TITLE: Draft Text of Recommendation H.263 Version 2 ("H.263+") for Decision

---

(#20 Complete Text with No Change Marks Version)

1. Introduction

Attached is the draft text of Version 2 of Recommendation H.263, which is informally known as "H.263+". It is submitted for decision at the January-February 1998 meeting of Study Group 16, after having reached determination at the March 1997 meeting. It contains a number of optional feature enhancements which are added to the core syntax and semantics of the original Version 1 of H.263 (the current published Version 1 is dated 3/96). In the process of adding these new features, the precise definition and requirements of the original Version 1 syntax and semantics have not been changed — only optional new features have been added. A description of the editorial status of this draft is given in Section 2 of this preamble, and an overview of its new features is provided in Section 3.

2. Draft status relative to the determined output of the Geneva SG16 meeting of 3/97:

This document has undergone review at two interim Experts Group meetings, and has required some technical as well as editorial modifications. Although some technical aspects have needed changes as a result of small errors found during the interim technical review and testing, the overall set of feature enhancements remains the same. No drafted annexes have been added or removed or had a change of title or purpose. One new non-normative Appendix has been added to provide guidance for users of the standard, to encourage the adoption of the various new optional features in a unified and interoperable way across the full spectrum of terminal and network types.

* Contact: Gary Sullivan
PictureTel Corp.
Andover, MA 01810

Tel.: +1.978.623.4324
Fax: +1.978.749.2804
E-mail: garys@pictel.com

3. Summary of New "H.263+" Extension Features

These revisions add optional features to Recommendation H.263 in order to broaden its range of useful application and to improve its compression performance. The additional optional feature set can be summarized in terms of the new types of pictures it can use, the new coding modes which can be applied to those pictures, and the definition of backward-compatible supplemental enhancement information which can be added into the video bitstream.

3.1. New types of pictures

3.1.1. Scalability pictures

Scalable video coding has potential for improving the delivery of video in error-prone, packet-loss-ridden, and heterogenous networks. It allows the video bitstream to be separated into multiple logical channels, so that some data can be lost or discarded without irreparable harm to the video representation. Three types of scalability pictures were added in this revision, one which provides temporal scalability and two which provide SNR or Spatial scalability:

1. B: A picture having two reference pictures, one of which temporally precedes the B picture and one of which is temporally subsequent to the B picture;
2. EI: A picture having a temporally simultaneous reference picture; and
3. EP: A picture having two reference pictures, one of which temporally precedes the EP picture and one of which is temporally simultaneous.

3.1.2. Improved PB frames

The existing Recommendation H.263 contains a special frame type called a "PB frame", which enables an increase in perceived frame rate with only a moderate increase in bit rate. However, recent investigations have indicated that the PB-frame as it exist is not sufficiently robust for continual use. Encoders wishing to use PB frames are limited in the types of prediction that a PB frame can use, which results in a lack of usefulness for PB frames in some situations. An improved, more robust type of PB frame has been added to enable heavier, higher-performance use of the PB frame design. It is a small modification of the existing PB frames mode.

3.1.3. Custom Source Formats

The existing Recommendation H.263 is very limited in the types of video input to which it can be applied. It allows only five video source formats defining the picture size, picture shape, and picture clock frequency. The new H.263+ feature set allows a wide range of optional custom source formats in order to make the standard apply to a much wider class of video scenes. These modifications help make the standard respond to the new world of resizable computer window-based displays, high refresh-rates, and wide format viewing screens.

3.2. New coding modes

This set of optional extensions of the H.263 video coding syntax also includes nine new coding modes that can be applied to the pictures in the bitstream:

1. Advanced INTRA Coding (AIC): A mode which improves the compression efficiency for INTRA macroblock encoding by using spatial prediction of DCT coefficient values;
2. Deblocking Filter (DF): A mode which reduces the amount of block artifacts in the final image by filtering across block boundaries using an adaptive filter;
3. Slice Structured (SS): A mode which allows a functional grouping of a number of macroblocks in the picture, enabling improved error resilience, improved transport over packet networks, and reduced delay;
4. Reference Picture Selection (RPS): A mode which improves error resilience by allowing a temporally previous reference picture to be selected which is not the most recent encoded picture that can be syntactically referenced;
5. Reference Picture Resampling (RPR): A mode which allows a resampling of a temporally previous reference picture prior to its use as a reference for encoding, enabling global motion compensation, predictive dynamic resolution conversion, predictive picture area alteration and registration, and special-effect warping;
6. Reduced-Resolution Update (RRU): A mode which allows an encoder to maintain a high frame rate during heavy motion by encoding a low-resolution update to a higher-resolution picture while maintaining high resolution in stationary areas;

7. Independent Segment Decoding (ISD): A mode which enhances error resilience by ensuring that corrupted data from some region of the picture cannot cause propagation of error into other regions;
8. Alternative INTER VLC (AIV): A mode which reduces the number of bits needed for encoding predictively-coded blocks when there are many large coefficients the block; and
9. Modified Quantization (MQ): A mode which improves the control of the bit rate by changing the method for controlling the quantizer step size on a macroblock basis, reduces the prevalence of chrominance artifacts by reducing the step size for chrominance quantization, increases the range of representable coefficient values for use with small quantizer step sizes, and increases error detection performance and reduces decoding complexity by prohibiting certain unreasonable coefficient representations.

3.3. Supplemental Enhancement Information

This revision also allows the addition of supplemental enhancement information to the video bitstream. Although this information does not affect the semantics for decoding the bitstream, it can enable enhanced features for systems which understand the optional additional information (while being discarded harmlessly by systems that do not understand it). This allows a variety of picture freeze and release commands, tagging information associated synchronously with the pictures in the bitstream for external use, and video transparency by use of chroma key information.

INTERNATIONAL TELECOMMUNICATION UNION

ITU-T DRAFT H.263

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(26 September, 1997)

TRANSMISSION OF NON-TELEPHONE SIGNALS

VIDEO CODING FOR LOW BIT RATE COMMUNICATION

ITU-T Recommendation H.263

(Previously "CCITT Recommendation")

FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union (ITU). The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, March 1-12, 1993).

ITU-T Recommendation H.263 was prepared by ITU-T Study Group 16 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the xx th of xx 1998.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

© ITU 199x

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

Page

1 Scope 1

2 References 1

3 Brief specification 1
- 3.1 Video input and output 1
- 3.2 Digital output and input 2
- 3.3 Sampling frequency 2
- 3.4 Source coding algorithm 2
- 3.5 Bit rate 5
- 3.6 Buffering 5
- 3.7 Symmetry of transmission 5
- 3.8 Error handling 5
- 3.9 Multipoint operation 5

4 Source coder 6
- 4.1 Source format 6
- 4.2 Video source coding algorithm 8
- 4.3 Coding control 11
- 4.4 Forced updating 11
- 4.5 Byte alignment of start codes 11

5 Syntax and semantics 12
- 5.1 Picture layer 12
- 5.2 Group of Blocks Layer 29
- 5.3 Macroblock Layer 30
- 5.4 Block Layer 37

6 Decoding process 41
- 6.1 Motion compensation 41
- 6.2. Coefficients decoding 42
- 6.3 Reconstruction of blocks 43

Annex A Inverse transform accuracy specification 45

Annex B Hypothetical Reference Decoder 47

Annex C Considerations for Multipoint 49
- C.1 Freeze picture request 49
- C.2 Fast update request 49
- C.3 Freeze picture release 49
- C.4 Continuous Presence Multipoint and Video Multiplexing (CPM) 49

Annex D Unrestricted Motion Vector mode 51
- D.1 Motion vectors over picture boundaries 51
- D.2 Extension of the motion vector range 51

Annex E Syntax-based Arithmetic Coding mode 55
- E.1 Introduction 55
- E.2 Specification of SAC encoder 55
- E.3 Specification of SAC decoder 56
- E.4 Syntax 57
- E.5 PSC_FIFO 57
- E.6 Header layer symbols 57
- E.7 Macroblock and Block layer symbols 58
- E.8 SAC Models 58

Annex F Advanced Prediction mode 62
- F.1 Introduction 62
- F.2 Four motion vectors per macroblock 62
- F.3 Overlapped motion compensation for luminance 63

Annex G PB-frames mode 66
- G.1 Introduction 66

G.2 PB-frames and INTRA blocks 66
G.3 Block Layer 67
G.4 Calculation of vectors for the B-picture in a PB-frame 67
G.5 Prediction of a B-block in a PB-frame 67

Annex H Forward Error Correction for coded video signal 70
H.1 Introduction 70
H.2 Error correction framing 70
H.3 Error correcting code 70
H.4 Relock time for error corrector framing 70

Annex I Advanced INTRA Coding mode 72
I.1 Introduction 72
I.2 Syntax 72
I.3 Decoding Process 73

Annex J Deblocking Filter mode 79
J.1 Introduction 79
J.2 Relation to UMV and AP modes (Annexes D and F) 79
J.3 Definition of the deblocking edge filter 80

Annex K Slice Structured mode 84
K.1 Introduction 84
K.2 Structure of slice layer 84

Annex L Supplemental Enhancement Information Specification 88
L.1 Introduction 88
L.2 PSUPP Format 88
L.3 Do Nothing 88
L.4 Full-Picture Freeze Request 89
L.5 Partial-Picture Freeze Request 89
L.6 Resizing Partial-Picture Freeze Request 89
L.7 Partial-Picture Freeze-Release Request 89
L.8 Full-Picture Snapshot Tag 90
L.9 Partial-Picture Snapshot Tag 90
L.10 Video Time Segment Start Tag 90
L.11 Video Time Segment End Tag 90
L.12 Progressive Refinement Segment Start Tag 90
L.13 Progressive Refinement Segment End Tag 90
L.14 Chroma Keying Information 90
L.15 Extended Function Type 92

Annex M Improved PB-frames mode 93
M.1 Introduction 93
M.2 B_{PB}-macroblock prediction modes 93
M.3 Calculation of vectors for bidirectional prediction of a the B-macroblock 94
M.4 MODB table 94

Annex N Reference Picture Selection mode 95
N.1 Introduction 95
N.2 Video source coding algorithm 95
N.3 Channel for Back-Channel messages 96
N.4 Syntax 97
N.5 Decoder Process 99

Annex O Temporal, SNR, and Spatial Scalability mode 101
O.1 Overview 101
O.2 Transmission order of pictures 104
O.3 Picture Layer Syntax 105
O.4 Macroblock Layer Syntax 106
O.5 Motion vector decoding 110
O.6 Interpolation filters 110

Annex P Reference Picture Resampling 112
P.1 Introduction 112
P.2 Syntax 114

P.3 Resampling algorithm 116
P.4 Example of implementation 118
P.5 Factor-of-4 resampling 120

Annex Q Reduced-Resolution Update mode 123
Q.1 Introduction 123
Q.2 Decoding Procedure 124
Q.3 Extension of Referenced Picture 125
Q.4 Reconstruction of motion vectors 126
Q.5 Enlarged overlapped motion compensation for luminance 128
Q.6 Upsampling of the reduced-resolution reconstructed prediction error 130
Q.7 Block boundary filter 132

Annex R Independent Segment Decoding mode 134
R.1 Introduction 134
R.2 Mode Operation 134
R.3 Constraints on Usage 134

Annex S Alternative INTER VLC mode 136
S.1 Introduction 136
S.2 Alternative INTER VLC for coefficients 136
S.3 Alternative INTER VLC for CBPY 136

Annex T Modified Quantization mode 137
T.1 Introduction 137
T.2 Modified DQUANT Update 137
T.3 Altered quantization step size for chrominance coefficients 138
T.4 Modified coefficient range 138
T.5 Usage Restrictions 139

Appendix I List of patent holders 140

Appendix II Error Tracking 140

Appendix III Recommended Optional Enhancement 141
III.1 Introduction 141
III.2 Levels of Preferred Mode Support 141
III.3 Picture Formats and Picture Clock Frequencies 143

SUMMARY

This Recommendation specifies a coded representation that can be used for compressing the moving picture component of audio-visual services at low bit rates. The basic configuration of the video source coding algorithm is based on Recommendation H.261 and is a hybrid of inter-picture prediction to utilize temporal redundancy and transform coding of the remaining signal to reduce spatial redundancy. The source coder can operate on five standardised video source formats: sub-QCIF, QCIF, CIF, 4CIF and 16CIF, and can also operate using a broad range of custom video formats.

The decoder has motion compensation capability, allowing optional incorporation of this technique in the coder. Half pixel precision is used for the motion compensation, as opposed to Recommendation H.261 where full pixel precision and a loopfilter are used. Variable length coding is used for the symbols to be transmitted.

In addition to the basic video source coding algorithm, sixteen negotiable coding options are included for improved compression performance and the support of additional capabilities. Additional supplemental information may also be included in the bitstream for enhanced display capability and for external usage.

VIDEO CODING FOR LOW BIT RATE COMMUNICATION

(Geneva, 1998)

1 Scope

This Recommendation specifies a coded representation that can be used for compressing the moving picture component of audio-visual services at low bit rates. The basic configuration of the video source coding algorithm is based on Recommendation H.261. Sixteen negotiable coding options are included for improved performance and increased functionality. This Recommendation contains Version 2 of Recommendation H.263, which is fully compatible with the original Recommendation, adding only optional features to the original Version 1 content of the Recommendation.

2 References

The following Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-R Recommendations is regularly published.

[1] ITU-R Recommendation BT.601-4 (199x): *Encoding Parameters of Digital Television for Studios*

Reference [1] is referenced herein to define the (Y, C_B, C_R) color space and its 8-bit integer representation for pictures used by video codecs designed according to this Recommendation. (Reference [1] is not used to define any other aspects of this Recommendation.)

The following additional ITU-T Recommendations are mentioned for illustration purposes in this text, however they do not constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

[2] ITU-T Recommendation H.223 (1996), *Multiplexing protocol for low bit rate multimedia communication.*

[3] ITU-T Recommendation H.242 (1996), System for establishing communication between audiovisual *terminals using digital channels up to 2 Mbit/s.*

[4] ITU-T Recommendation H.245 (1996), *Control protocol for multimedia communication.*

[5] ITU-T Recommendation H.261 (1993), Video codec for audiovisual services at p *x 64 kbit/s.*

[6] ITU-T Rec. H.262 (1995) | *ISO/IEC 13818-2:1995, Information technology* – Generic coding of moving pictures and associated audio *information: video.*

[7] ITU-T Recommendation H.324 (1996), *Terminal for low bit rate multimedia communication.*

3 Brief specification

An outline block diagram of the codec is given in Figure 1.

3.1 Video input and output

To permit a single Recommendation to cover use in and between regions using 625- and 525-line television standards, the standardised source formats on which the source coder operates are based on a Common Intermediate Format (CIF). It is also possible using external negotiation (for example Recommendation H.245), to enable the use of a wide range of optional custom source formats. The standards of the input and output television signals, which may, for example, be composite or component, analogue or digital and the methods of performing any necessary conversion to and from the source coding format are not subject to recommendation.

3.2 Digital output and input

The video coder provides a self-contained digital bitstream which may be combined with other multi-facility signals (for example as defined in Recommendation H.223). The video decoder performs the reverse process.

3.3 Sampling frequency

Pictures are sampled at an integer multiple of the video line rate. This sampling clock and the digital network clock are asynchronous.

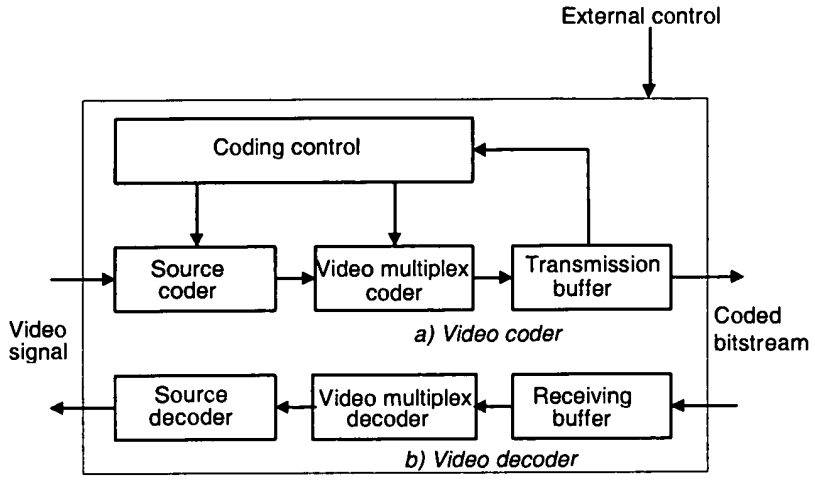


FIGURE 1/H.263

Outline block diagram of the video codec

3.4 Source coding algorithm

A hybrid of inter-picture prediction to utilize temporal redundancy and transform coding of the remaining signal to reduce spatial redundancy is adopted. The decoder has motion compensation capability, allowing optional incorporation of this technique in the coder. Half pixel precision is used for the motion compensation, as opposed to Recommendation H.261 where full pixel precision and a loopfilter are used. Variable length coding is used for the symbols to be transmitted.

In addition to the core H.263 coding algorithm, sixteen negotiable coding options can be used, either together or separately (subject to certain restrictions). Additional supplemental information may also be included in the bitstream for enhanced display capability and for external usage. A forward error correction method for application to the resulting video bitstream is provided for use when necessary. The negotiable coding options, forward error correction, and supplemental information usage are described in the subsequent subsections.

3.4.1 Continuous Presence Multipoint and Video Multiplex mode

In this optional mode up to four separate video "Sub-Bitstreams" can be sent within the same video channel. This feature is designed for use in continuous presence multipoint application or other situations in which separate logical channels are not available but the use of multiple video bitstreams is desired (see also Annex C).

3.4.2 Unrestricted Motion Vector mode

In this optional mode motion vectors are allowed to point outside the picture. The edge pixels are used as prediction for the "non-existing" pixels. With this mode a significant gain is achieved if there is movement across the edges of the picture, especially for the smaller picture formats (see also Annex D). Additionally, this mode includes an extension of the motion vector range so that larger motion vectors can be used. This is especially useful in case of camera movement and large picture formats.

3.4.3 Syntax-based Arithmetic Coding mode

In this optional mode arithmetic coding is used instead of variable length coding. The SNR and reconstructed pictures will be the same, but significantly fewer bits will be produced (see also Annex E).

3.4.4 Advanced Prediction mode

In this optional mode Overlapped Block Motion Compensation (OBMC) is used for the luminance part of P-pictures (see also Annex F). Four 8 × 8 vectors instead of one 16 × 16 vector are used for some of the macroblocks in the picture. The encoder has to decide which type of vectors to use. Four vectors use more bits, but give better prediction. The use of this mode generally gives a considerable improvement. A subjective gain is achieved because OBMC results in less blocking artifacts.

3.4.5 PB-frames mode

A PB-frame consists of two pictures being coded as one unit. The name PB comes from the name of picture types in Recommendation H.262 where there are P-pictures and B-pictures. Thus a PB-frame consists of one P-picture which is predicted from the previous decoded P-picture and one B-picture which is predicted from both the previous decoded P-picture and the P-picture currently being decoded. The name B-picture was chosen because parts of B-pictures may be bidirectionally predicted from the past and future pictures. With this coding option, the picture rate can be increased considerably without substantially increasing the bitrate (see also Annex G). However, an Improved PB frames mode is also provided (see also Annex M). The original PB frames mode is retained herein only for purposes of compatibility with systems made prior to the adoption of the Improved PB frames mode.

3.4.6 Forward Error Correction

A forward error correction method is specified for use when necessary for the protection of the video bitstream for use when appropriate. The method provided for forward error correction is the same BCH code method specified also in ITU-T Recommendation H.261 (see also Annex H).

3.4.7 Advanced INTRA Coding mode

In this optional mode, INTRA blocks are first predicted from neighboring INTRA blocks prior to coding (see also Annex I). Separate variable length code (VLC) tables are defined for the INTRA blocks. The technique is applied to INTRA-macroblocks within INTRA pictures and to INTRA-macroblocks within INTER pictures. This mode significantly improves the compression performance over the INTRA coding of the core H.263 syntax.

3.4.8 Deblocking Filter mode

In this optional mode, a filter is applied across the 8x8 block edge boundaries of decoded I- and P-pictures to reduce blocking artifacts (see also Annex J). The purpose of the filter is to mitigate the occurrence of block edge artifacts in the decoded picture. The filter affects the picture that is used for the prediction of subsequent pictures and thus lies within the motion prediction loop.

3.4.9 Slice Structured mode

In this optional mode, a "slice" layer is substituted for the GOB layer of the bitstream syntax (see also Annex K). The purposes of this mode are to provide enhanced error resilience capability, to make the bitstream more amenable to use with an underlying packet transport delivery, and to minimize video delay. A slice is similar to a GOB, in that it is a layer of the syntax that lies between the picture layer and the macroblock layer. However, the use of a slice layer allows a flexible partitioning of the picture, in contrast with the fixed partitioning and fixed transmission order required by the GOB structure.

3.4.10 Supplemental Enhancement Information

Additional supplemental information may be included in the bitstream to signal enhanced display capability or to provide information for external usage (see also Annex L). This supplemental information can be used to signal a full-picture or partial-picture freeze or freeze-release request with or without resizing, can also be used to tag particular pictures or sequences of pictures within the video stream for external usage, and can be used to convey chroma key information for video compositing. The supplemental information may be present in the bitstream even though the decoder may not be capable of providing the enhanced capability to use it, or even to properly interpret it — simply discarding the supplemental information is allowable by decoders unless a requirement to provide the requested capability has been negotiated by external means.

3.4.11 Improved PB-frames mode

This optional mode represents an improvement compared to the PB-frames mode option (see also annexes G and M). The main difference between the two modes is that in the Improved PB-frames mode, each B-block may be forward predicted using a separate motion vector or backward predicted with a zero vector. This significantly improves coding efficiency in situations in which downscaled P-vectors are not good candidates for B-prediction. The backward prediction is particularly useful when there is a scene cut between the previous P-frame and the BPB-frame.

3.4.12 Reference Picture Selection mode

An optional mode is provided which improves the performance of real-time video communication over error-prone channel by allowing temporal prediction from pictures other than the most recently-sent reference picture (see also Annex N). This mode can be used with back-channel status messages which are sent back to the encoder to inform it about whether its bitstream is being properly received. In error-prone channel environments, this mode allows the encoder to optimize its video encoding for the conditions of the channel.

3.4.13 Temporal, SNR and Spatial Scalability mode

In this optional mode there is support for temporal, SNR, and spatial scalability (see also Annex O). Scalability implies that a bitstream is composed of a base layer and one or more associated enhancement layers. The base layer is a separately decodable bitstream. The enhancement layers can be decoded in conjunction with the base layer to increase perceived quality by either increasing the picture rate, increasing the picture quality, or increasing the picture size. SNR scalability refers to enhancement information to increase the picture quality without increasing picture resolution. Spatial scalability refers to enhancement information to increase the picture quality by increasing picture resolution either horizontally, vertically, or both. There is also support for temporal scalability by the use of B pictures. A B picture is a scalability enhancement containing pictures that can be bi-directionally predicted from two pictures in a reference layer, one temporally previous to the current picture and one temporally subsequent. B pictures allows enhancement layer information to be used to increase perceived quality by increasing the picture rate of the displayed enhanced video sequence. This mode can be useful for heterogenous networks with varying bandwidth capacity and also in conjunction with error correction schemes.

3.4.14 Reference Picture Resampling mode

A syntax is provided for supporting an optional mode for which the reference picture used for video image prediction is processed by a resampling operation prior to its use in forming a predictor for the current input picture (see also Annex P). This allows efficient dynamic selection of an appropriate image resolution for video encoding, and can also support picture warping for use as a global motion compensator or special-effect generator.

3.4.15 Reduced-Resolution Update mode

An optional mode is provided which allows reduced resolution updates to a reference picture having a higher resolution (see also Annex Q). This mode is expected to be used when encoding a highly active scene, and allows an encoder to increase the picture rate at which moving parts of a scene can be represented, while maintaining a higher resolution representation in more static areas of the scene.

3.4.16 Independent Segment Decoding mode

An optional mode is provided which allows a picture to be constructed without any data dependencies which cross GOB or multi-GOB video picture segments or slice boundaries (see also Annex R). This mode provides error robustness by preventing the propagation of erroneous data across the boundaries of the video picture segment areas.

3.4.17 Alternative INTER VLC mode

An optional mode is provided which improves the efficiency of INTER picture coding when significant changes are evident in the picture (see also Annex S). This efficiency improvement is obtained by allowing a VLC code originally designed for INTRA pictures to be used for some INTER picture coefficients as well.

3.4.18 Modified Quantization mode

An optional mode is provided which improves the bit-rate control ability for encoding, reduces chrominance quantization error, expands the range of representable DCT coefficients, and places certain restrictions on coefficient values (see also Annex T). This mode modifies the semantics of the differential quantization step size parameter of the bitstream by broadening the range of step size changes that can be specified. It also reduces the quantization step size used for chrominance data. The range of DCT coefficient levels is broadened to ensure that any possible coefficient value can be encoded to within the accuracy allowed by the step size. Certain restrictions

are also placed on coefficients in this mode to increase error detection performance and minimize decoder complexity.

3.5 Bit rate

The transmission clock is provided externally. The video bit rate may be variable. In this Recommendation no constraints on the video bit rate are given; constraints will be given by the terminal or the network.

3.6 Buffering

The encoder shall control its output bitstream to comply with the requirements of the hypothetical reference decoder defined in Annex B. Video data shall be provided on every valid clock cycle. This can be ensured by MCBPC stuffing (see Tables 7 and 8) or, when forward error correction is used, also by forward error correction stuffing frames (see Annex H).

The number of bits created by coding any single picture shall not exceed a maximum value specified by the parameter BPPmaxKb which is measured in units of 1024 bits. The minimum allowable value of the BPPmaxKb parameter depends on the largest picture size that has been negotiated for use in the bitstream (see Table 1). The picture size is measured as the picture width times height for the luminance (Y) component, measured in pixels. An encoder may use a larger value for BPPmaxKb than as specified in Table 1, provided the larger value is first negotiated by external means, for example Recommendation H.245.

When the Temporal, SNR, and Spatial Scalability mode (Annex O) is employed, the number of bits sent for each picture in each enhancement layer shall not exceed the maximum value specified by BPPmaxKb.

TABLE 1/H.263

Minimum BPPmaxKb for different source picture formats

Y Picture Size in Pixels	Minimum BPPmaxKb
up to 25,344 (or QCIF)	64
25,360 to 101,376 (or CIF)	256
101,392 to 405,504 (or 4CIF)	512
405,520 and above	1024

3.7 Symmetry of transmission

The codec may be used for bidirectional or unidirectional visual communication.

3.8 Error handling

Error handling should be provided by external means (for example Recommendation H.223). If it is not provided by external means (for example in Recommendation H.221) the optional error correction code and framing as described in Annex H can be used.

A decoder can send a command to encode one or more GOBs (or slices if Annex K is employed) of its next picture in INTRA mode with coding parameters such as to avoid buffer overflow. A decoder can also send a command to transmit only non-empty GOB headers if the Slice Structured mode (see Annex K) is not in use. The transmission method for these signals is by external means (for example Recommendation H.245).

3.9 Multipoint operation

Features necessary to support switched multipoint operation are included in Annex C.

4 Source coder

4.1 Source format

The source coder operates on non-interlaced pictures having a source format which is defined in terms of 1) the picture format, as determined by the number of pixels per line, the number of lines per picture, and the pixel aspect ratio, and 2) the timing between pictures, as determined by the picture clock frequency (PCF). For example, the common intermediate format (CIF) has 352 pixels per line, 288 lines, a pixel aspect ratio of 12:11, and a picture clock frequency of 30 000/1001 pictures per second.

The source coder operates on non-interlaced pictures occurring at a picture clock frequency (PCF) of 30 000/1001 (approximately 29.97) times per second, termed the CIF PCF. It is also possible to negotiate the use of an optional custom PCF by external means. A custom PCF is given by 1 800 000 / (clock divisor * clock conversion factor) where clock divisor can have values of 1 through 127 and clock conversion factor can either be 1000 or 1001.□The tolerance on the picture clock frequency is ± 50 ppm.

Pictures are coded as luminance and two colour difference components (Y, C_B and C_R). These components and the codes representing their sampled values are as defined in ITU-R Recommendation BT.601-4.

- Black = 16;
- White = 235;
- Zero colour difference = 128;
- Peak colour difference = 16 and 240.

These values are nominal ones and the coding algorithm functions with input values of 1 through to 254.

There are five standardised picture formats: sub-QCIF, QCIF, CIF, 4CIF and 16CIF. It is also possible to negotiate a custom picture format. For all of these picture formats, the luminance sampling structure is dx pixels per line, dy lines per picture in an orthogonal arrangement. Sampling of each of the two colour difference components is at dx/2 pixels per line, dy/2 lines per picture, orthogonal. The values of dx, dy, dx/2 and dy/2 are given in Table 2 for each of the standardised picture formats.

TABLE 2/H.263

Number of pixels per line and number of lines for each of the standardised H.263 picture formats

Picture Format	number of pixels for luminance (dx)	number of lines for luminance (dy)	number of pixels for chrominance (dx/2)	number of lines for chrominance (dy/2)
sub-QCIF	128	96	64	48
QCIF	176	144	88	72
CIF	352	288	176	144
4CIF	704	576	352	288
16CIF	1408	1152	704	576

For all picture formats, colour difference samples are sited such that their block boundaries coincide with luminance block boundaries as shown in Figure 2. The pixel aspect ratio is the same for each of the standardised picture formats and is the same as defined for QCIF and CIF in Recommendation H.261: (288/3):(352/4), which simplifies to 12:11 in relatively prime numbers. The picture area covered by all of the standardised picture formats except the sub-QCIF picture format has an aspect ratio of 4:3.

Custom picture formats can have a custom pixel aspect ratio as described in Table 3, if the custom pixel aspect ratio use is first negotiated by external means. Custom picture formats can have any number of lines and any number of pixels per line, provided that the number of lines is divisible by four and is in the range [4,...,1152], and provided that the number of pixels per line is also divisible by four and is in the range [4,...2048]. For picture formats having a width or height that is not divisible by 16, the picture is decoded in the same manner as if the width or height had the next larger size that would be divisible by 16 and then the picture is cropped at the right and the bottom to the proper width and height for display purposes only.

TABLE 3/H.263

Custom Pixel Aspect Ratios

Pixel Aspect Ratio	Pixel Width : Pixel Height
Square	1:1
CIF	12:11
525-type for 4:3 picture	10:11
CIF for 16:9 picture	16:11
525-type for 16:9 picture	40:33
Extended PAR	m:n, m and n are relatively prime

All decoders and encoders shall be able to operate using the CIF picture clock frequency. Some decoders and encoders may also support custom picture clock frequencies. All decoders shall be able to operate using the sub-QCIF picture format. All decoders shall also be able to operate using the QCIF picture format. Some decoders may also operate with CIF, 4CIF or 16CIF, or custom picture formats. Encoders shall be able to operate with one of the picture formats sub-QCIF and QCIF. The encoders determine which of these two formats are used, and are not obliged to be able to operate with both. Some encoders can also operate with CIF, 4CIF, 16CIF, or custom picture formats. Which optional formats and which picture clock frequencies can be handled by the decoder is signalled by external means, for example Recommendation H.245. For a complete overview of possible picture formats and video coding algorithms, refer to the terminal description, for example Recommendation H.324.

NOTE – For CIF, the number of pixels per line is compatible for practical purposes with sampling the active portions of the luminance and colour difference signals from 525- or 625-line sources at 6.75 and 3.375 MHz respectively. These frequencies have a simple relationship to those in ITU-R Recommendation BT.601-4.

Means shall be provided to restrict the maximum picture rate of encoders by having a minimum number of non-transmitted pictures between transmitted ones. Selection of this minimum number shall be by external means (for example, Recommendation H.245). For the calculation of the minimum number of non-transmitted pictures in PB-frames mode, the P-picture and the B-picture of a PB-frames unit are taken as two separate pictures.

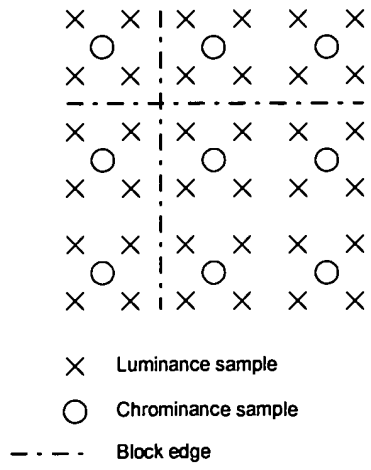
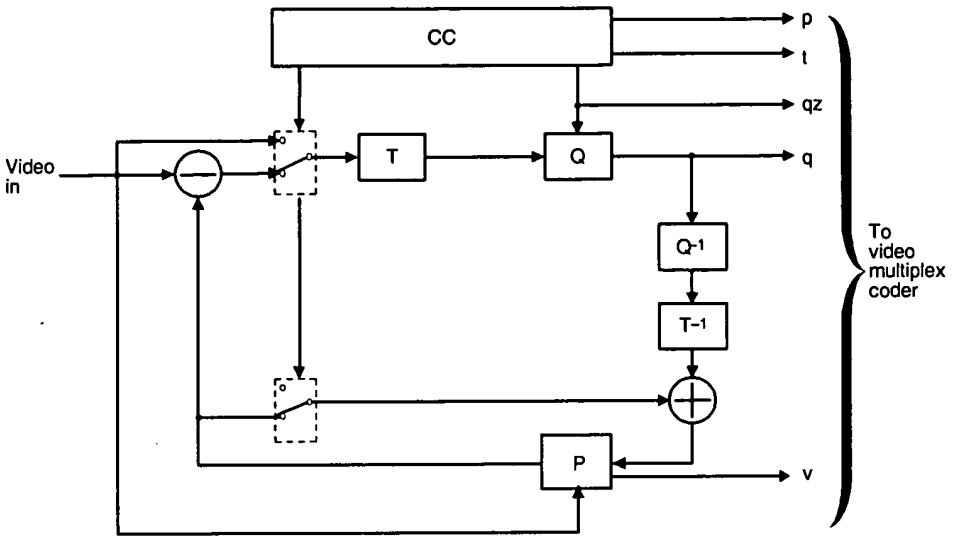


FIGURE 2/H.263

Positioning of luminance and chrominance samples

4.2 Video source coding algorithm

The source coder is shown in generalized form in Figure 3. The main elements are prediction, block transformation and quantization.



T Transform
Q Quantizer
P Picture Memory with motion compensated variable delay

CC Coding control
p Flag for INTRA/INTER
t Flag for transmitted or not
qz Quantizer indication
q Quantizing index for transform coefficients
v Motion vector

FIGURE 3/H.263

Source coder

4.2.1 GOBs, slices, macroblocks and blocks

Each picture is divided either into groups of blocks (GOBs) or into slices.

A group of blocks (GOB) comprises of up to k*16 lines, where k depends on the number of lines in the picture format and depends on whether the optional Reduced-Resolution Update mode is in use (see Annex Q). The dependencies are shown in Table 4. If the number of lines is less than or equal to 400 and the optional Reduced-Resolution Update mode is not in use, then k = 1. If the number of lines is less than or equal to 800 and the optional Reduced-Resolution Update mode is in use or the number of lines is more than 400, then k=2. If the number of lines is more than 800, then k=4. When using custom picture sizes, the number of lines in the last (bottom-most) GOB may be less than k*16 if the number of lines in the picture is not divisible by k*16. However, every GOB in each of the standardized picture formats has k*16 lines, as the number of lines in each standardized picture format is an integer multiple of k*16. Thus, for example, if the optional Reduced-Resolution mode is not in use, the number of GOBs per picture is 6 for sub-QCIF, 9 for QCIF, and 18 for CIF, 4CIF and 16CIF. The GOB numbering is done by use of vertical scan of the GOBs, starting with the upper GOB (number 0) and ending with the bottom-most GOB. An example of the arrangement of GOBs in a picture is given for the CIF picture format in Figure 4. Data for each GOB consists of a GOB header (may be empty) followed by data for macroblocks. Data for GOBs is transmitted per GOB in increasing GOB number.

TABLE 4/H.263

Parameter k for GOB size definition

Number of Lines dy	Value of k when not in RRU mode	Value of k when in RRU mode
4,...,400	1	2
404,...,800	2	2
804,...,1152	4	4

The Slice Structured mode is described in Annex K. Slices are similar to GOBs in that they are a multi-macroblock layer of the syntax, but slices have a more flexible shape and usage than GOBs, and slices may appear in the bitstream in any order, under certain conditions.

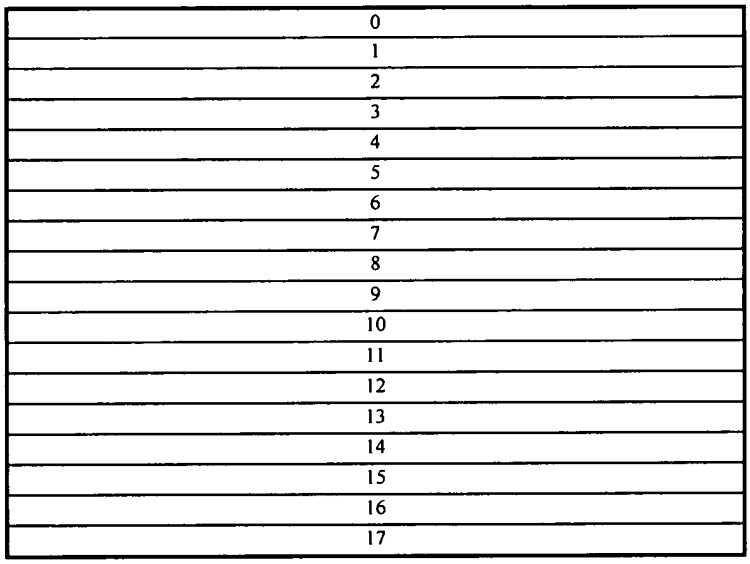


FIGURE 4/H.263

Arrangement of Groups of Blocks in a CIF picture

Each GOB is divided into macroblocks. The macroblock structure depends on whether the optional Reduced-Resolution Update (RRU) mode is in use (see Annex Q). Unless in RRU mode, each macroblock relates to 16 pixels by 16 lines of Y and the spatially corresponding 8 pixels by 8 lines of C_B and C_R. Further, a macroblock consists of four luminance blocks and the two spatially corresponding colour difference blocks as shown in Figure 5. Each luminance or chrominance block thus relates to 8 pixels by 8 lines of Y, C_B or C_R. Unless in RRU mode, a GOB comprises one macroblock row for sub-QCIF, QCIF and CIF, two macroblock rows for 4CIF and four macroblock rows for 16CIF.

When in RRU mode, a macroblock relates to 32 pixels by 32 lines of Y and the spatially corresponding 16 pixels by 16 lines of C_B and C_R, and each luminance or chrominance block relates to 16 pixels by 16 lines of Y, C_B or C_R. Furthermore, a GOB comprises one macroblock row for CIF and 4CIF, and two macroblock rows for 16CIF.

The macroblock numbering is done by use of horizontal scan of the macroblock rows from left to right, starting with the upper macroblock row and ending with the lower macroblock row. Data for the macroblocks is transmitted per macroblock in increasing macroblock number. Data for the blocks is transmitted per block in increasing block number (see Figure 5).

The criteria for choice of mode and transmitting a block are not subject to recommendation and may be varied dynamically as part of the coding control strategy. Transmitted blocks are transformed and resulting coefficients are quantized and entropy coded.

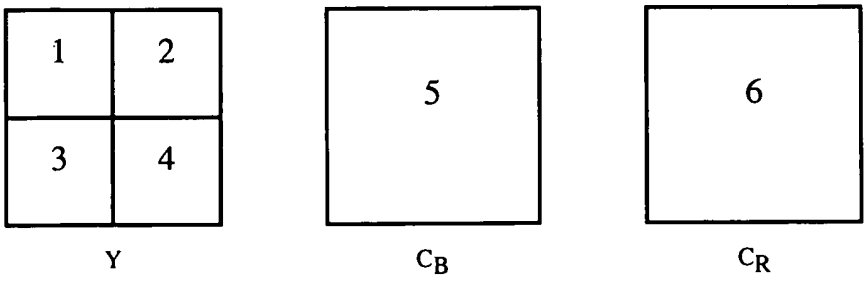


FIGURE 5/H.263

Arrangement of blocks in a macroblock

4.2.2 Prediction

The primary form of prediction is inter-picture and may be augmented by motion compensation (see 4.2.3). The coding mode in which temporal prediction is applied is called INTER; the coding mode is called INTRA if no temporal prediction is applied. The INTRA coding mode can be signalled at the picture level (INTRA for I-pictures or INTER for P-pictures) or at the macroblock level in P-pictures. In the optional PB-frames mode, B-pictures are always coded in INTER mode. The B-pictures are partly predicted bidirectionally (refer to Annex G).

In total, H.263 has seven basic picture types (of which only the first two are mandatory) which are defined primarily in terms of their prediction structure:

1. INTRA: A picture having no reference picture(s) for prediction (also called an I picture);
2. INTER: A picture using a temporally-previous reference picture (also called a P picture);
3. PB: A frame representing two pictures and having a temporally-previous reference picture (see Annex G);
4. Improved PB: A frame functionally similar but normally better than a PB frame (see Annex M);
5. B: A picture having two reference pictures, one of which temporally precedes the B picture and one of which temporally succeeds the B picture and has the same picture size (see Annex O);
6. EI: A picture having a temporally simultaneous reference picture which has either the same or a smaller picture size (see Annex O); and
7. EP: A picture having two reference pictures, one of which temporally precedes the EP picture and one of which is temporally simultaneous and has either the same or a smaller picture size (see Annex O).

As used herein, a "reference" or "anchor" picture is a picture that contains data which can be used by reference as a basis for the decoding of another picture. This use by reference is also known as "prediction", although it sometimes may actually indicate use in a reverse-temporal direction.

4.2.3 Motion compensation

The decoder will accept one vector per macroblock or, if the Advanced Prediction mode or the Deblocking Filter mode is used, one or four vectors per macroblock (see Annexes F and J). If the PB-frames mode is used, one additional delta vector can be transmitted per macroblock for adaptation of the motion vectors for prediction of the B-macroblock. Similarly, an Improved PB-frame macroblock (see Annex M) can include an additional forward motion vector. B-picture macroblocks (see Annex O) can be transmitted together with a forward and a backward motion vector, and EP-pictures can be transmitted with a forward motion vector.

Both horizontal and vertical components of the motion vectors have integer or half integer values. In the default prediction mode, these values are restricted to the range [–16, 15.5] (this is also valid for the forward and backward motion vector components for B-pictures).

In the Unrestricted Motion Vector mode however, the maximum range for vector components is increased. If PLUSPTYPE is absent, the range is [–31.5, 31.5], with the restriction that only values that are within a range of [–16, 15.5] around the predictor for each motion vector component can be reached if the predictor is in the range [–15.5, 16]. If PLUSPTYPE is absent and the predictor is outside [–15.5, 16], all values within the range [–31.5, 31.5] with the same sign as the predictor plus the zero value can be reached. If PLUSPTYPE is present, the motion vector values are less restricted (see also Annex D).

In the Reduced-Resolution Update mode, the motion vector range is enlarged to approximately double size, and each vector component is restricted to have only a half-integer or zero value. Therefore the range of each motion vector component is [-31.5, 30.5] in the default Reduced-Resolution Update mode (see Annex Q) and a larger range if the Unrestricted Motion Vector mode is also used (see also Annex D).

A positive value of the horizontal or vertical component of the motion vector signifies that the prediction is formed from pixels in the referenced picture which are spatially to the right or below the pixels being predicted.

Motion vectors are restricted such that all pixels referenced by them are within the coded picture area, except when the Unrestricted Motion Vector mode, the Advanced Prediction mode, the Advanced Prediction mode, or the Deblocking Filter mode is used (see Annexes D, F, and J), or in B and EP pictures of the Temporal, SNR, and Spatial Scalability mode (see Annex O).

4.2.4 Quantization

Unless the optional Advanced INTRA Coding mode or Modified Quantization mode is in use, the number of quantizers is 1 for the first coefficient of INTRA blocks and 31 for all other coefficients. Within a macroblock the same quantizer is used for all coefficients except the first one of INTRA blocks. The decision levels are not defined. The first coefficient of INTRA blocks is nominally the transform dc value uniformly quantized with a step size of 8. Each of the other 31 quantizers use equally spaced reconstruction levels with a central dead-zone around zero and with a step size of an even value in the range 2 to 62. For the exact formulas, refer to 6.2. For quantization using the Advanced INTRA Coding mode, refer to Annex I. For quantization using the Modified Quantization mode, refer to Annex T.

NOTE – For the smaller quantization step sizes, the full dynamic range of the transform coefficients cannot be represented unless the optional Modified Quantization mode is in use.

4.3 Coding control

Several parameters may be varied to control the rate of generation of coded video data. These include processing prior to the source coder, the quantizer, block significance criterion and temporal subsampling. The proportions of such measures in the overall control strategy are not subject to recommendation.

When invoked, temporal subsampling is performed by discarding complete pictures.

A decoder can signal its preference for a certain tradeoff between spatial and temporal resolution of the video signal. The encoder shall signal its default tradeoff at the beginning of the call and shall indicate whether it is capable of responding to decoder requests to change this tradeoff. The transmission method for these signals is by external means (for example, Recommendation H.245).

4.4 Forced updating

This function is achieved by forcing the use of the INTRA mode of the coding algorithm. The update pattern is not defined. To control the accumulation of inverse transform mismatch error, each macroblock shall be coded in INTRA mode at least once every 132 times when coefficients are transmitted for this macroblock in P-pictures. A similar requirement applies when using optional EP pictures (see Annex O), for which each macroblock shall be coded in an INTRA or upward mode at least once every 132 times when coefficients are transmitted for the macroblock.

4.5 Byte alignment of start codes

Byte alignment of start codes is achieved by inserting a stuffing codeword consisting of less than 8 zero-bits before the start code such that the first bit of the start code is the first (most significant) bit of a byte. A start code is therefore byte aligned if the position of its most significant bit is a multiple of 8-bits from the first bit in the H.263 bitstream. All picture, slice, and EOSBS start codes shall be byte aligned, and GOB and EOS start codes may be byte aligned.

NOTES

1 The number of bits spent for a certain picture is variable but always a multiple of 8 bits.

2 H.324 requires H.263 encoders to align picture start codes with the start of logical information units passed to the Adaptation Layer (AL_SDU's).

5 Syntax and semantics

The video syntax is arranged in a hierarchical structure with four primary layers. From top to bottom the layers are:

- Picture;
- Group of Blocks, or slice, or video picture segment;
- Macroblock;
- Block.

The syntax diagram is shown in Figure 6. A guide for interpretation of the diagram consists of the following:

1. Arrow paths show the possible flows of syntax elements. Any syntax element which has zero length is considered absent for arrow path diagramming purposes (thus, for example, there is an arrow path bypassing PSTUF despite the mandatory nature the PSTUF field since the length of the PSTUF field may be zero).
2. Abbreviations and semantics for each syntax element are as defined in later clauses.
3. The set of syntax elements and arrow paths shown using thick solid lines denotes the syntax flow of the "baseline" mode of operation, without the use of any optional enhancements. (This syntax was also present in the Version 1 of this Recommendation, and remains unchanged in any way.)
4. The set of syntax elements and arrow paths shown using thick dotted lines denotes the additional elements in the syntax flow of the optional enhancements which have been present in both Version 1 and Version 2 of this Recommendation. (This syntax remains unchanged in any way.)
5. The set of syntax elements and arrow paths shown using thin solid lines denotes the additional new elements in the syntax flow of the optional enhancements which are specific to the additional optional features added in Version 2. (This syntax was not present in Version 1.)
6. Syntax element fields shown with square-edged boundaries indicate fixed-length fields, and those with rounded boundaries indicate variable-length fields. One syntax element (DQUANT) is shown with both types of boundaries, because can have either a variable or fixed length.
7. A fixed-length field is defined to be a field for which the length of the field is not dependent on the data in the content of the field itself. The length of this field is either always the same, or is determined by the prior data in the syntax flow.
8. The term "layer" is used to refer to any part of the syntax which can be understood and diagrammed as a distinct entity.

Unless specified otherwise, the most significant bit is transmitted first. This is bit 1 and is the left most bit in the code tables in this Recommendation. Unless specified otherwise, all unused or spare bits are set to "1". Spare bits shall not be used until their functions are specified by the ITU.

5.1 Picture layer

Data for each picture consists of a picture header followed by data for Group of Blocks or for slices, eventually followed by an optional end-of-sequence code and stuffing bits. The structure is shown in Figure 7 for pictures that do not include the optional PLUSPTYPE data field. PSBI is only present if indicated by CPM. TR_B and DBQUANT are only present if PTYPE indicates use of the 'PB-frame' mode (unless the PLUSPTYPE field is present and the use of DBQUANT is indicated therein).

The optional PLUSPTYPE data field is present when so indicated in bits 6-8 of PTYPE. When present, an additional set of data is included in the bitstream, which immediately follows PTYPE and precedes PQUANT. In addition, the CPM and PSBI fields are moved forward in the picture header when PLUSPTYPE is present, so that they appear immediately after PLUSPTYPE rather than being located after PQUANT. The format for the additional data following PLUSPTYPE is shown in Figure 8. All fields in this additional picture header data after PLUSPTYPE are optional, depending on whether their presence is indicated in PLUSPTYPE. When the Slice Structured mode (see Annex K) is in use, slices are substituted for GOBs in the location shown in Figure 7.

Combinations of PSUPP and PEI may not be present, and may be repeated when present. EOS and EOSBS+ESBI may not be present, while ESTUF may be present only if EOS or EOSBS is present. EOS shall not be repeated unless at least one picture start code appears between each pair of EOS codes. Picture headers for dropped pictures are not transmitted.

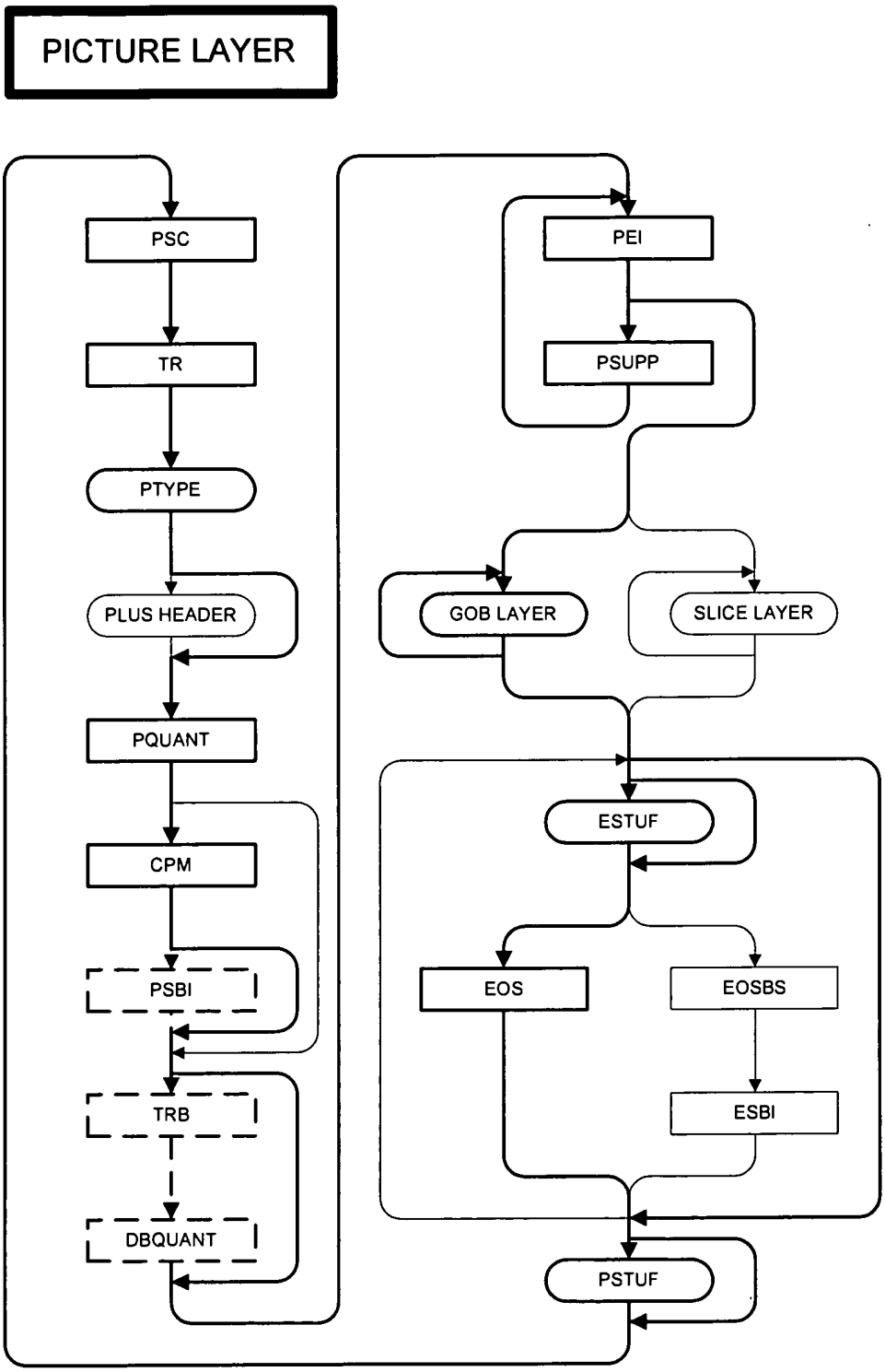


FIGURE 6/H.263

Syntax diagram for the video bitstream (part 1 of 7 parts)

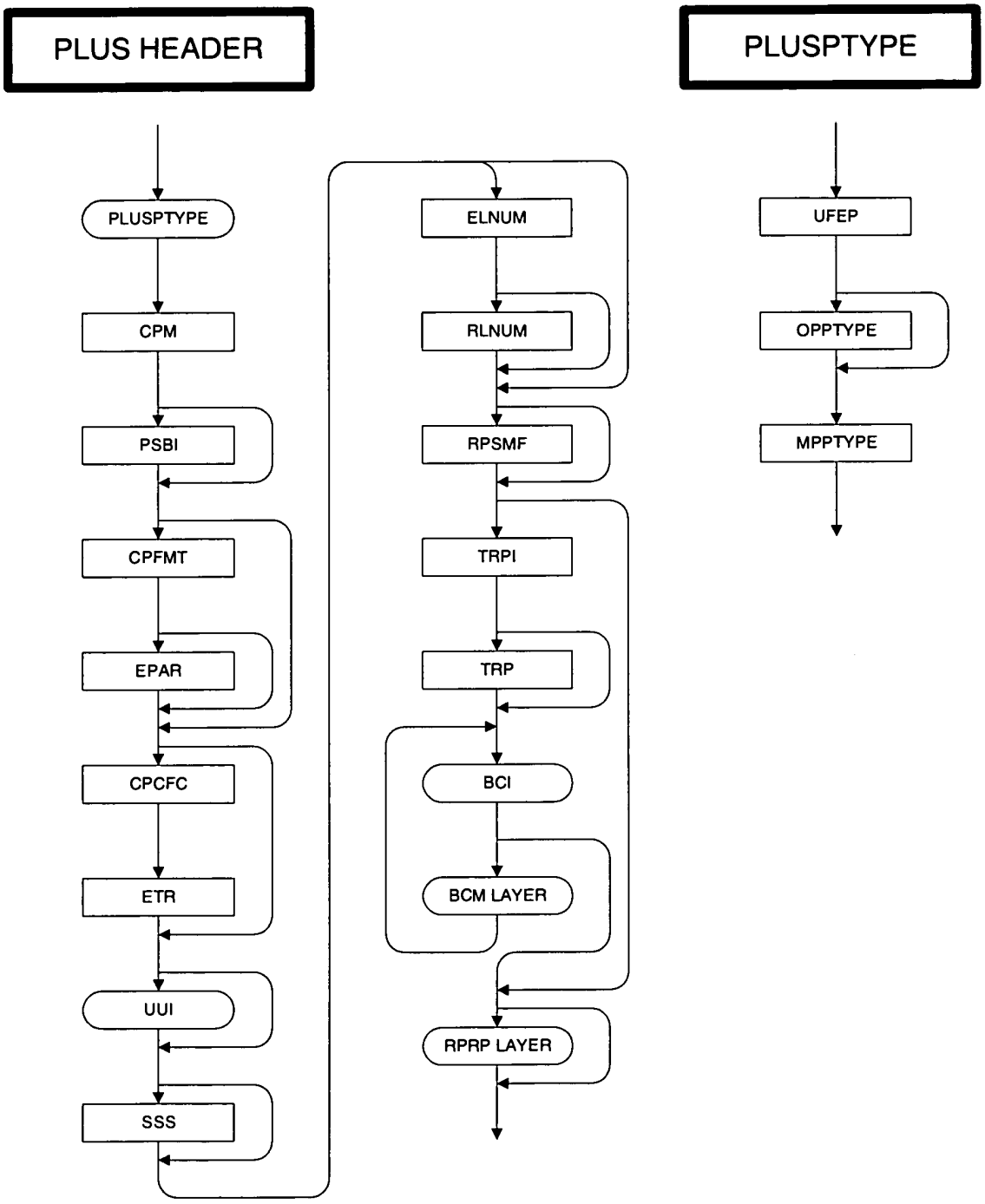


FIGURE 6/H.263 (continued)

Syntax diagram for the video bitstream (part 2 of 7 parts)

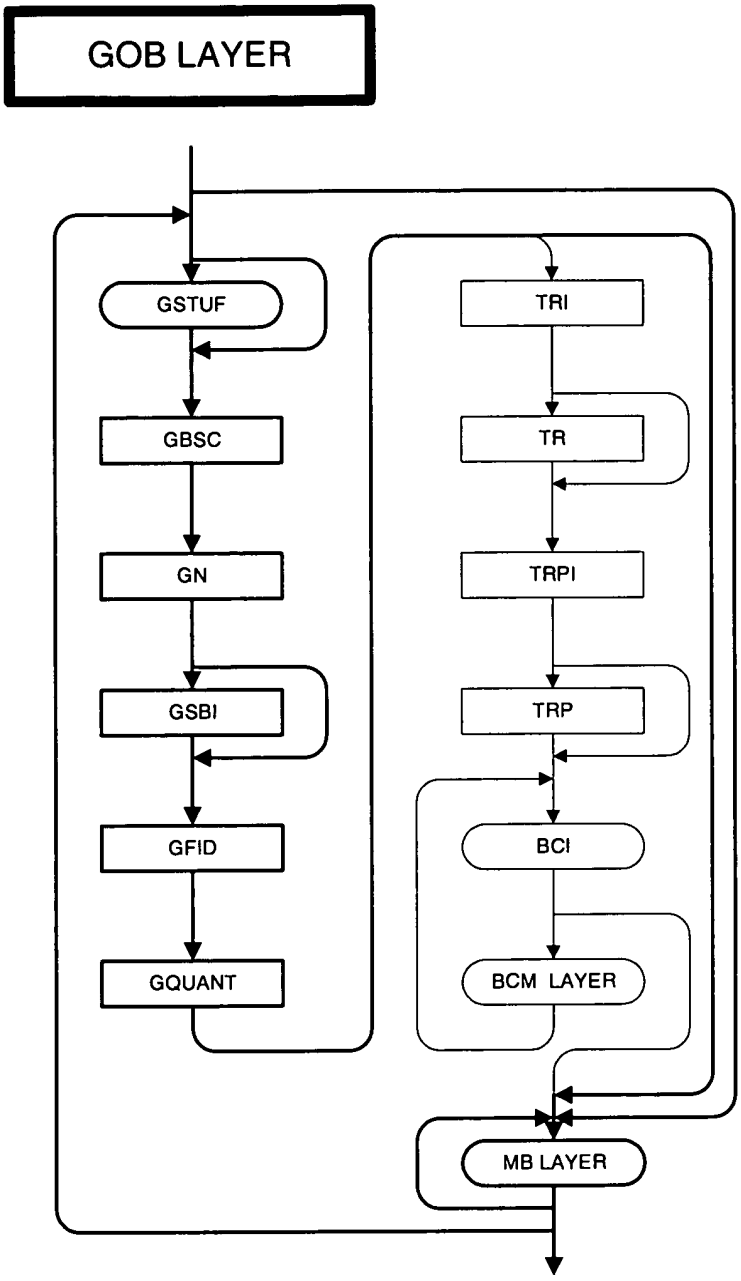


FIGURE 6/H.263 (continued)

Syntax diagram for the video bitstream (part 3 of 7 parts)

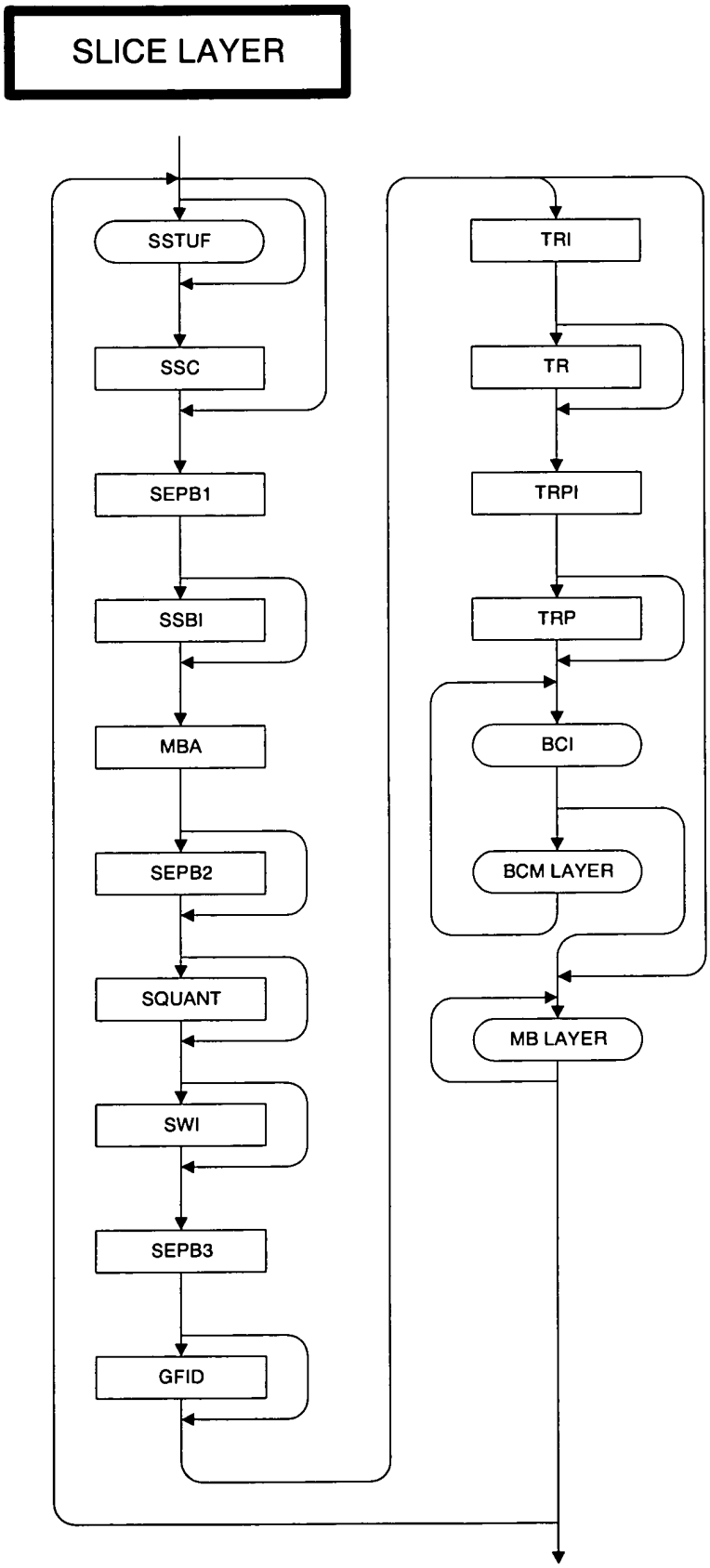


FIGURE 6/H.263 (continued)

Syntax diagram for the video bitstream (part 4 of 7 parts)

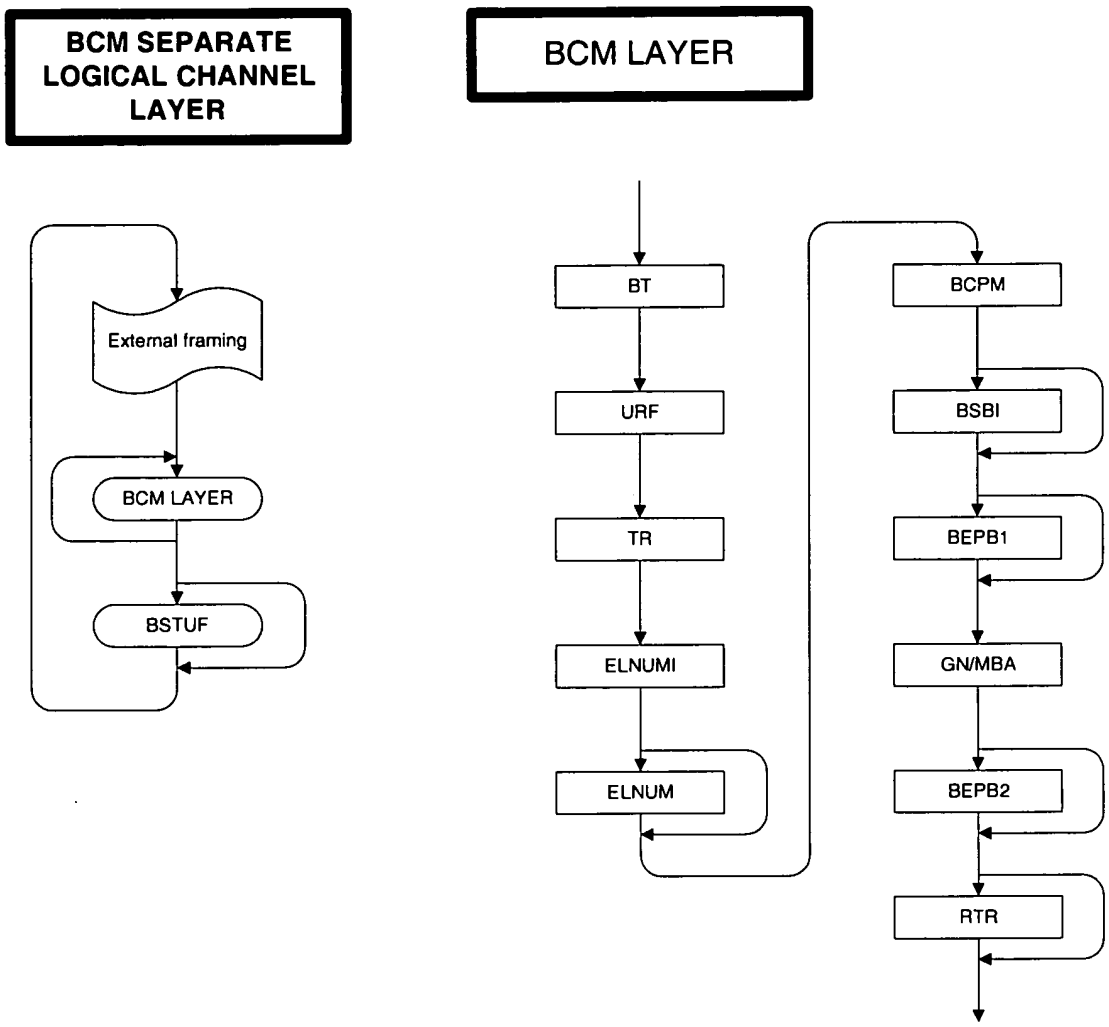


FIGURE 6/H.263 (continued)

Syntax diagram for the video bitstream (part 5 of 7 parts)

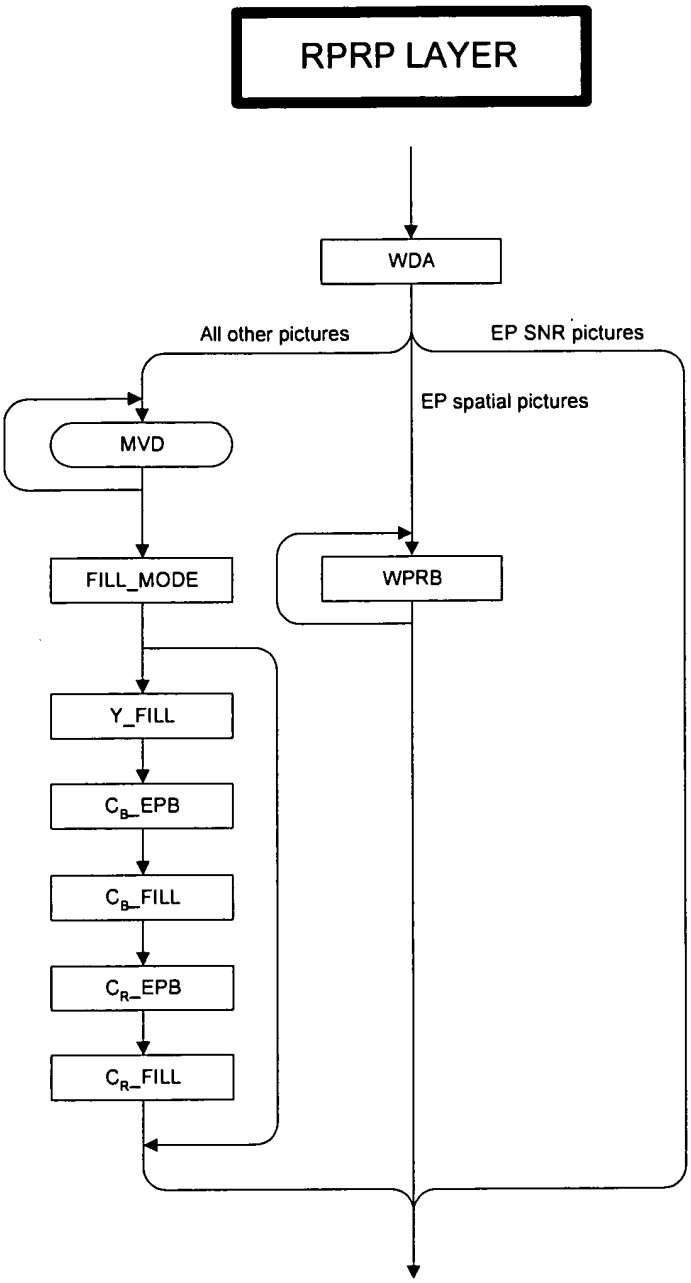


FIGURE 6/H.263 (continued)

Syntax diagram for the video bitstream (part 6 of 7 parts)

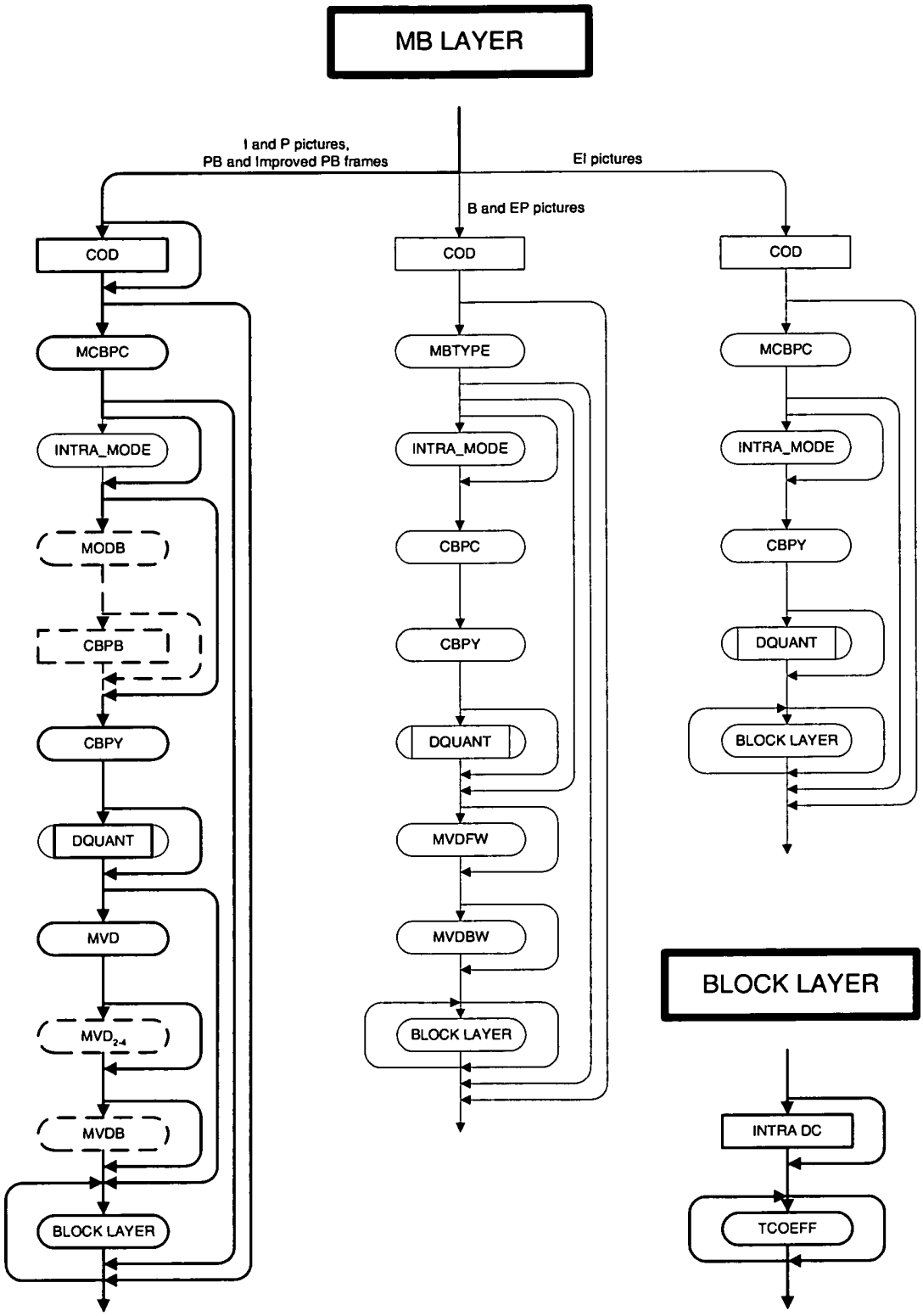


FIGURE 6/H.263 (continued)

Syntax diagram for the video bitstream (part 7 of 7 parts)

FIGURE 7/H.263

Structure of picture layer
(without optional PLUSPTYPE-related fields)

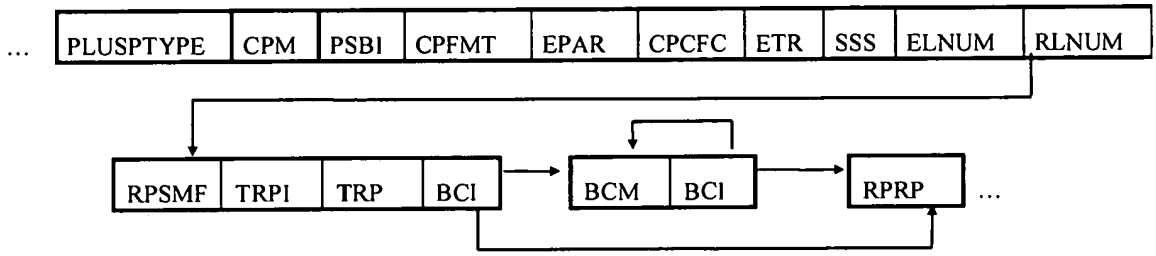


FIGURE 8/H.263

Structure of optional PLUSPTYPE-related fields
(located immediately after PTYPE when present)

5.1.1 Picture Start Code (PSC) (22 bits)

PSC is a word of 22 bits. Its value is 0000 0000 0000 0000 1 00000. All picture start codes shall be byte aligned. This shall be achieved by inserting PSTUF bits as necessary before the start code such that the first bit of the start code is the first (most significant) bit of a byte.

5.1.2 Temporal Reference (TR) (8 bits)

The value of TR is formed by incrementing its value in the temporally-previous reference picture header by one plus the number of skipped or non-reference pictures at the picture clock frequency since the previously transmitted one. The interpretation of TR depends on the active picture clock frequency. Under the standard CIF picture clock frequency, TR is a 8-bit number which can have 256 possible values. The arithmetic is performed with only the eight LSBs. If a custom picture clock frequency is signalled in use, Extended TR in section 5.1.8 and TR form a 10-bit number where TR stores the eight least significant bits (LSBs) and ETR stores the 2 most significant bits (MSBs). The arithmetic in this case is performed with the ten LSBs. In the optional PB-frames or Improved PB-frames mode, TR only addresses P-pictures; for the temporal reference for the B-picture part of PB or Improved PB frames refer to section 5.1.22.

5.1.3 Type Information (PTYPE) (Variable Length)

Information about the complete picture:

- Bit 1: Always "1", in order to avoid start code emulation.
- Bit 2: Always "0", for distinction with Recommendation H.261.
- Bit 3: Split screen indicator, "0" off, "1" on.
- Bit 4: Document camera indicator, "0" off, "1" on.
- Bit 5: Full Picture Freeze Release, "0" off, "1" on.

– Bits 6-8: Source Format, "000" forbidden, "001" sub-QCIF, "010" QCIF, "011" CIF, "100" 4CIF, "101" 16CIF, "110" reserved, "111" extended PTYPE,

If bits 6-8 are not equal to "111", which indicates an extended PTYPE (PLUSPTYPE) the following five bits are also present in PTYPE:

– Bit 9: Picture Coding Type, "0" INTRA (I-picture), "1" INTER (P-picture),

– Bit 10: Optional Unrestricted Motion Vector mode (see Annex D), "0" off, "1" on.

– Bit 11: Optional Syntax-based Arithmetic Coding mode (see Annex E), "0" off, "1" on.

– Bit 12: Optional Advanced Prediction mode (see Annex F), "0" off, "1" on.

– Bit 13: Optional PB-frames mode (see Annex G), "0" normal I- or P-picture, "1" PB-frame.

Split screen indicator is a signal that indicates that the upper and lower half of the decoded picture could be displayed side by side. This bit has no direct effect on the encoding or decoding of the picture.

Full Picture Freeze Release is a signal from an encoder which responds to a request for packet retransmission (if not acknowledged) or fast update request (see also Annex C) or picture freeze request (see also Annex L) and allows a decoder to exit from its freeze picture mode and display the decoded picture in the normal manner.

If bits 6-8 indicate a different source format than in the previous picture header, the current picture shall be an I-picture, unless an extended PTYPE is indicated in bits 6-8 and the capability to use the optional Reference Picture Resampling mode (see Annex P) has been negotiated externally (for example Recommendation H.245).

Bits 10-13 refer to optional modes that are only used after negotiation between encoder and decoder (see also the Annexes D, E, F and G, respectively). If bit 9 is set to "0", bit 13 shall be set to "0" as well.

Bits 6-8 shall not have a value of '111' which indicates the presence of an extended PTYPE (PLUSPTYPE) unless the capability has been negotiated externally (for example Recommendation H.245) to allow the use of a custom source format or one or more of the other optional modes available only by the use of an extended PTYPE (see Annexes I through K and M through T). Whenever bit 6-8 do not have a value of '111', all of the additional modes available only by the use of an extended PTYPE shall be considered to have been set to an "off" state and shall be inferred to remain "off" unless explicitly switched on later in the bitstream.

5.1.4 Plus PTYPE (PLUSPTYPE) (Variable Length)

A codeword of 12 or 30 bits which is present only if the presence of extended PTYPE is indicated in bits 6-8 of PTYPE.

PLUSPTYPE is comprised of up to three subfields, UFEP, OPPTYPE, and MPPTYPE. OPPTYPE is present only if UFEP has a particular value.

5.1.4.1 Update Full Extended PTYPE (UFEP) (3 bits)

A fixed length codeword of 3 bits which is present only if "extended PTYPE" is indicated in PTYPE bits 6-8. When set to "000", it indicates that only those extended PTYPE fields which need to be signaled in every picture header (MPPTYPE) are included in the current picture header. When set to "001", indicates that all extended PTYPE fields are included in the current picture header. If the picture type is INTRA or EI, this field shall be set to "001".

In addition, if PLUSPTYPE is present in each of a continuing sequence of pictures, this field shall be set to "001" at least as often as specified by a five second or five picture timeout period, whichever allows a larger interval of time. More specifically, the timeout period requires UFEP="001" to appear in the PLUSPTYPE field (if PLUSPTYPE is present in every intervening picture) of the first picture header with temporal reference indicating a time interval greater than or equal to five seconds since the last occurrence of UFEP="001", or of the fifth picture after the last occurrence of UFEP="001" (whichever requirement allows a longer period of time as measured by temporal reference).

Encoders should set UFEP to "001" more often in error-prone environments. Values of UFEP other than '000' and '001' are reserved.

5.1.4.2 The optional part of PLUSPTYPE (OPPTYPE) (18 bits)

If UFEP is "001", then the following bits are present in PLUSPTYPE:

Bit 1-3 Source Format, "000" reserved, "001" sub-QCIF, "010" QCIF, "011" CIF, "100" 4CIF, "101" 16CIF, "110" custom source format, "111" reserved,

Bit 4 Optional Custom PCF, "0" CIF PCF, "1" custom PCF.,

Bit 5 Optional Unrestricted Motion Vector (UMV) mode (see Annex D), "0" off, "1" on,

Bit 6 Optional Syntax-based Arithmetic Coding (SAC) mode (see Annex E), "0" off, "1" on,

Bit 7 Optional Advanced Prediction (AP) mode (see Annex F), "0" off, "1" on,

Bit 8 Optional Advanced INTRA Coding (AIC) mode (see Annex I), "0" off, "1" on,

Bit 9 Optional Deblocking Filter (DF) mode (see Annex J), "0" off, "1" on,

Bit 10 Optional Slice Structured (SS) mode (see Annex K), "0" off, "1" on,

Bit 11 Optional Reference Picture Selection (RPS) mode (see Annex N), "0" off, "1" on,

Bit 12 Optional Independent Segment Decoding (ISD) mode (see Annex R),"0" off, "1" on,

Bit 13 Optional Alternative INTER VLC (AIV) mode (see Annex S), "0" off, "1" on,

Bit 14 Optional Modified Quantization (MQ) mode (see Annex T), "0" off, "1" on,

Bit 15 Equal to "1" to prevent start code emulation,

Bit 16 Reserved, shall be equal to "0",

Bit 17 Reserved, shall be equal to "0".

Bit 18 Reserved, shall be equal to "0".

5.1.4.3. The mandatory part of PLUSPTYPE when PLUSPTYPE present (MPPTYPE) (9 bits)

Regardless of the value of UFEP, the following 9 bits are also present in PLUSPTYPE:

Bit 1-3 Picture Type Code:
"000" I-picture (INTRA),
"001" P-picture (INTER),
"010" Improved PB frame (see Annex M),
"011" B-picture (see Annex O),
"100" EI-picture (see Annex O),
"101" EP-picture (see Annex O),
"110" reserved,
"111" reserved

Bit 4 Optional Reference Picture Resampling (RPR) mode (see Annex P), "0" off, "1" on,

Bit 5 Optional Reduced-Resolution Update (RRU) mode (see Annex Q), "0" off, "1" on,

Bit 6 Rounding Type (RTYPE) (see section 6.1.2),

Bit 7 Reserved, shall be equal to "0",

Bit 8 Reserved, shall be equal to "0",

Bit 9 Equal to "1" to prevent start code emulation.

The encoder should control the rounding type so that P pictures, Improved PB frames, and EP pictures have different values for Bit 6 (Rounding Type for P-Pictures) from their reference pictures for motion compensation. Bit 6 can have an arbitrary value if the reference picture is an I or an EI picture. Bit 6 can be set to "1" only when Bits 1-3 indicate a P Picture, Improved PB frame, or EP Picture. For other types of pictures, this bit shall always be set to "0".

5.1.4.4 The semantics of PLUSPTYPE

The mandatory part of PLUSPTYPE consists of features which are likely to be changed on a picture-by-picture basis. Primarily, these are the bits that indicate the picture type among I, P, Improved PB, B, EI, and EP. (Note that the PB-frames mode of Annex G cannot be used with PLUSPTYPE present—the Improved PB frames mode of Annex M should be used instead.) However, they also include indications for the use of the RPR and RRU modes, as these may change from picture to picture as well.

Features that are likely to remain in use rather than being changed from one picture to another (except in obvious ways as discussed in section 5.4.4.5), have been placed in the optional part of PLUSPTYPE. When UFEP is 000,

the missing mode information is inferred from the type of picture and from the mode information sent in a previous PLUSPTYPE with UFEP equal to 001.

If PLUSPTYPE is present but UFEP is 000, then

1. For a P picture or Improved PB frame (see Annex M), the pixel aspect ratio, picture width, and picture height are unchanged from those of the reference picture.

2. For a temporal-scalability B picture (see Annex O) in an enhancement layer, the reference layer number (RLNUM) is the same as the enhancement layer number (ELNUM) if the last picture sent in the enhancement layer was an EI or EP picture. If the last picture sent in the enhancement layer was a B picture, the reference layer number is the same as the reference layer number of the last B picture. The pixel aspect ratio, picture width and picture height are unchanged from those of the temporally subsequent reference layer picture.

   Note that if temporally surrounding EI or EP pictures exist in the same enhancement layer as the B picture, RLNUM (explicit or implied) shall always be equal to ELNUM. Note also that the pixel aspect ratio, picture width and picture height of a B picture (explicit or implied) shall always be equal to those of its temporally subsequent reference layer picture.

3. For a SNR/Spatial scalability EP picture (see Annex O), the pixel aspect ratio, picture width and picture height are unchanged from those of the temporally previous reference picture in the same enhancement layer.

5.1.4.5 Mode restrictions for certain picture types and mode inference rules

Certain modes do not apply to certain types of pictures. Particularly, these restrictions apply:

1. The following modes do not apply within I (INTRA) pictures: Unrestricted Motion Vector (see Annex D), Advanced Prediction (see Annex F), Alternative INTER VLC (see Annex S), Reference Picture Resampling (see Annex P), and Reduced Resolution Update (see Annex Q).

2. The following modes do not apply within B pictures (see Annex O): Syntax-based Arithmetic Coding (see Annex E), Deblocking Filter (see Annex J), and Advanced Prediction (see Annex F).

3. The following modes do not apply within EI pictures (see Annex O): Unrestricted Motion Vector (see Annex D), Syntax-based Arithmetic Coding (see Annex E), Advanced Prediction (see Annex F), Reference Picture Resampling (see Annex P), Reduced Resolution Update (see Annex Q), and Alternative INTER VLC (see Annex S).

4. The following modes do not apply within EP pictures (see Annex O): Syntax-based Arithmetic Coding (see Annex E) and Advanced Prediction (see Annex F).

One or more of the modes listed in the above four-item list may have a mode flag having a value "1" in the optional part of PLUSPTYPE within a picture of a type that is prohibited for that mode (types I, B, EI, or EP). This condition is allowed and shall be interpreted subject to the mode inference rules which follow in the next paragraph.

Mode states are subject to the following mode inference rules:

1. Once a mode flag has been set to "1" in the optional part of PLUSPTYPE, the current picture and each subsequent picture in the bitstream shall be assigned a state of "on" for that mode.

2. An inferred state of "off" shall be assigned to any mode which does not apply within a picture having the current picture type code. However, each subsequent picture in the bitstream shall have an inferred state of "on" for that mode (unless this too results in an obvious conflict—which shall be resolved in the same way). In the case of layered scalable bitstreams (see Annex O), the mode state shall be inferred only from within the same layer of the bitstream.

3. The inference of state shall continue until another picture containing the optional part of PLUSPTYPE is sent or until a picture which does not contain PLUSPTYPE at all is sent, whichever comes first. If a new picture containing the optional part of PLUSPTYPE is sent, the state sent in the new message shall override the old state. If a picture is sent which does not contain PLUSPTYPE (a picture in which bit 6-8 of PTYPE is not '111'), a state of "off" shall be assigned to all modes not explicitly set to "on" in the PTYPE field, and all modes shall continue to have an inferred state of "off" until a new picture containing the optional part of PLUSPTYPE is sent.

4. Two modes do not require mode state inference, since the mode flags for these modes appear in the mandatory part of PLUSPTYPE. These are the Reference Picture Resampling mode (Annex P) and the Reduced Resolution Update mode (Annex Q). The mode flag for either these modes shall not be set unless the current picture allows the use of the mode. For example, the Reduced Resolution Update mode bit shall not be set in an INTRA picture.

5.1.4.6 Mode interaction restrictions

Certain modes cannot be used in combination with certain other modes.

1. The Syntax-based Arithmetic Coding mode (see Annex E) shall not be used with the Alternative INTER VLC mode (see Annex S) or the Modified Quantization mode (see Annex T).
2. If PLUSPTYPE is present, the Unrestricted Motion Vector mode (see Annex D) shall not be used with the Syntax-based Arithmetic Coding mode (see Annex E).
3. The Independent Segment Decoding mode (see Annex R) shall not be used with the Reference Picture Resampling mode (see Annex P).
4. The Independent Segment Decoding mode (see Annex R) shall not be used with the Slice Structured mode without the simultaneous use of the Rectangular Slice submode of the Slice Structured mode (see Annex K).

5.1.4.7 The picture header location of CPM (1 bit) and PSBI (2 bits)

The location of the CPM and PSBI fields in the picture header depends on whether or not PLUSPTYPE is present (see sections 5.1.20 and 5.1.21). If PLUSPTYPE is present, then CPM follows immediately after PLUSPTYPE in the picture header. If PLUSPTYPE is not present, then CPM follows immediately after PQUANT in the picture header. PSBI always follows immediately after CPM (if CPM="1").

5.1.5 Custom Picture Format (CPFMT) (23 bits)

A fixed length codeword of 23 bits that is present only if the use of a custom picture format is signaled in PLUSPTYPE and UFEP is '001'. When present, CPFMT consists of:

Bit 1-4 Pixel Aspect Ratio Code: A 4-bit index to the PAR value in Table 5. For extended PAR, the exact pixel aspect ratio shall be specified in EPAR (section 5.1.6).,

Bit 5-13 Picture Width Indication: Range [0,...,511]; Number of pixels per line = (PWI+1)*4,

Bit 14 Equal to "1" to prevent start code emulation,

Bit 15-23 Picture Height Indication: Range [1,...,288]; Number of lines = PHI*4.

TABLE 5/H.263

PAR Code Definition

PAR Code	Pixel Aspect Ratios
0000	forbidden
0001	1:1 (Square)
0010	12:11 (CIF for 4:3 picture)
0011	10:11 (525-type for 4:3 picture)
0100	16:11 (CIF stretched for 16:9 picture)
0101	40:33 (525-type stretched for 16:9 picture)
0110-1110	reserved
1111	extended PAR

5.1.6 Extended Pixel Aspect Ratio (EPAR) (16 bits)

A fixed length codeword of 16 bits that is present only if CPFMT is present and extended PAR is indicated therein. When present, EPAR consists of:

Bit 1-8 PAR Width: "0" is forbidden. The natural binary representation of the PAR width,
Bit 9-16 PAR Height: "0" is forbidden. The natural binary representation of the PAR height.

The PAR Width and PAR Height shall be relatively prime.

5.1.7 Custom Picture Clock Frequency Code (CPCFC) (8 bits)

A fixed length codeword of 8 bits that is present only if PLUSPTYPE is present and UFEP is 001 and a custom picture clock frequency is signalled in PLUSPTYPE. When present, CPCFC consists of:

Bit 1 Clock Conversion Code: "0" indicates a clock conversion factor of 1000 and "1" indicates 1001.

Bit 2-8 Clock Divisor: "0" is forbidden. The natural binary representation of the value of the clock divisor.

The custom picture clock frequency is given by 1 800 000/(clock divisor * clock conversion factor) Hz.

The temporal reference counter shall count in units of the inverse of the picture clock frequency, in seconds. When the PCF changes from that specified for the previous picture, the temporal reference for the current picture is measured in terms of the prior PCF, so that the new PCF takes effect only for the temporal reference interpretation of future pictures.

5.1.8 Extended Temporal Reference (ETR) (2 bits)

A fixed length codeword of 2 bits which is present only if a custom picture clock frequency is in use (regardless of the value of UFEP). It is the two MSBs of the 10-bit number defined in section 5.1.2.

5.1.9 Unlimited Unrestricted Motion Vectors Indicator (UUI) (Variable Length)

A variable length codeword of 1 or 2 bits that is present only if the optional Unrestricted Motion Vector mode is indicated in PLUSPTYPE and UFEP is 001. When UUI is present it indicates the effective limitation of the range of the motion vectors being used.

UUI = "1" : The motion vector range is limited according to Tables D.1 and D.2.

UUI = "01" : The motion vector range is not limited except by the picture size.

5.1.10 Slice Structured submode bits (SSS) (2 bits)

A fixed length codeword of 2 bits which is present only if the optional Slice Structured mode (see Annex K) is indicated in PLUSPTYPE and UFEP is 001. If the Slice Structured mode is in use but UFEP is not 001, the last values sent for SSS shall remain in effect.

Bit 1 Rectangular Slices, "0" indicates free-running slices, "1" indicates rectangular slices.

Bit 2 Arbitrary Slice Ordering, "0" indicates sequential order, "1" indicates arbitrary order.

5.1.11 Enhancement layer number (ELNUM) (4 bits)

A fixed length codeword of 4 bits which is present only if the optional Temporal, SNR, and Spatial Scalability mode is in use (regardless of the value of UFEP). The particular enhancement layer is identified by an enhancement layer number, ELNUM. Picture correspondence between layers is achieved via the temporal reference. Picture size is either indicated within each enhancement layer using the existing source format fields or is inferred by the relationship to the reference layer. The first enhancement layer above the base layer is designated as Enhancement Layer Number 2, and the base layer has number 1.

5.1.12 Reference layer number (RLNUM) (4 bits)

A fixed length codeword of 4 bits which is present only if the optional Temporal, SNR, and Spatial Scalability mode is in use (see Annex O) and UFEP is 001. The layer number for the pictures used as reference anchors is identified by a reference layer number (RLNUM). Time correspondence between layers is achieved via the temporal reference.

Note that for B pictures in an enhancement layer having temporally surrounding EI or EP pictures which are present in the same enhancement layer, RLNUM shall be equal to ELNUM (see Annex O).

5.1.13 Reference Picture Selection Mode Flags (RPSMF) (3 bits)

A fixed length codeword of 3 bits that is present only if the Reference Picture Selection mode is in use and UFEP is 001. When present, RPSMF indicates which type of back-channel messages are needed by the encoder. If the Reference Picture Selection mode is in use but RPSMF is not present, the last value of RPSMF that was sent shall remain in effect.

100: neither ACK nor NACK signals needed
101: need ACK signals to be returned

110: need NACK signals to be returned
111: need both ACK and NACK signals to be returned
000-011: reserved

5.1.14 Temporal reference for prediction indication (TRPI) (1 bit)

A fixed length codeword of 1 bit that is present only if the optional Reference Picture Selection mode is in use (regardless of the value of UFEP). When present, TRPI indicates the presence of the following TRP field.

0: TRP field is not present.
1: TRP field is present.

TRPI shall be 0 whenever the picture header indicates an I or EI picture.

5.1.15 Temporal Reference for Prediction (TRP) (10 bits)

When present (as indicated in TRPI), TRP indicates the Temporal Reference which is used for prediction of the encoding, except for in the case of B pictures. For B pictures, the picture having the temporal reference TRP is used for the prediction in the forward direction. (Prediction in the reverse-temporal direction always uses the immediately temporally-subsequent picture.) TRP is a ten bit number. If a custom picture clock frequency was not in use for the reference picture, the two MSBs of TR are zero and the LSBs contain the eight bit TR found in the picture header of the reference picture. If a custom picture clock frequency was in use for the reference picture, TR is a ten bit number consisting of the concatenation of ETR and TR from the reference picture header.

5.1.16 Back-Channel message Indication (BCI) (Variable Length)

A variable length field of one or two bits that is present only if the optional Reference Picture Selection mode is in use. When set to "1", this signals the presence of the following optional video back-channel message (BCM) field. "01" indicates the absence or the end of the video back-channel message field. Combinations of BCM and BCI may not be present, and may be repeated when present. BCI shall be set to "01" if the videomux submode of the optional Reference Picture Selection mode is not in use.

5.1.17 Back-Channel Message (BCM) (Variable Length)

The Back-Channel message with syntax as specified in section N.4.2, which is present only if the preceding BCI field is present and is set to "1".

5.1.18 Reference Picture Resampling Parameters (RPRP) (Variable Length)

A variable length field that is present only if the optional Reference Picture Resampling mode bit is set in PLUSPTYPE. This field carries the parameters of the Reference Picture Resampling mode (see Annex P). Note that the Reference Picture Resampling mode can also be invoked implicitly by the occurrence of a picture header for an INTER coded picture having a picture size which differs from that of the previous encoded picture, in which case the RPRP field is not present and the Reference Picture Resampling mode bit is not set.

5.1.19 Quantizer Information (PQUANT) (5 bits)

A fixed length codeword of 5 bits which indicates the quantizer QUANT to be used for the picture until updated by any subsequent GQUANT or DQUANT. The codewords are the natural binary representations of the values of QUANT which, being half the step sizes, range from 1 to 31.

5.1.20 Continuous Presence Multipoint and Video Multiplex (CPM) (1 bit)

A codeword of 1 bit that signals the use of the optional Continuous Presence Multipoint and Video Multiplex mode (CPM); "0" is off, "1" is on. For the use of CPM, refer to Annex C. CPM follows immediately after PLUSPTYPE if PLUSPTYPE is present, but follows PQUANT in the picture header if PLUSPTYPE is not present.

5.1.21 Picture Sub-Bitstream Indicator (PSBI) (2 bits)

A fixed length codeword of 2 bits that is only present if Continuous Presence Multipoint and Video Multiplex mode is indicated by CPM. The codewords are the natural binary representation of the sub-bitstream number for the picture header and all following information until the next Picture or GOB start code (see also Annex C). PSBI follows immediately after CPM if CPM is "1" (the location of CPM and PSBI in the picture header depend on whether PLUSPTYPE is present).

5.1.22 Temporal Reference for B-pictures in PB-frames (TR_B) (3/5 bits)

TR_B is present if PTYPE or PLUSPTYPE indicates 'PB-frame' or 'Improved PB-frame' (see also Annexes G and M) and indicates the number of non-transmitted or non-reference pictures (at 29.97 Hz or the custom picture clock frequency indicated in CPCFC) since the last P- or I-picture or P part of a PB or Improved PB frame and before the B-picture part of the PB- or Improved PB-frame. The codeword is the natural binary representation of the number of non-transmitted pictures plus one. It is 3-bits long for standard CIF picture clock frequency and is extended to five bits when a custom picture clock frequency is in use. The maximum number of non-transmitted pictures is 6 for the standard CIF picture clock frequency and 30 when a custom picture clock frequency is used.

5.1.23 Quantization information for B-pictures in PB frames (DBQUANT) (2 bits)

DBQUANT is present if PTYPE or PLUSPTYPE indicates 'PB-frame' or 'Improved PB-frame' (see also Annexes G and M). In the decoding process a quantization parameter QUANT is obtained for each macroblock. With PB-frames QUANT is used for the P-block, while for the B-block a different quantization parameter BQUANT is used. QUANT ranges from 1 to 31. DBQUANT indicates the relation between QUANT and BQUANT as defined in Table 6. In this table, "/" means division by truncation. BQUANT ranges from 1 to 31; if the value for BQUANT resulting from Table 6 is greater than 31, it is clipped to 31.

TABLE 6/H.263

DBQUANT codes and relation between QUANT and BQUANT

DBQUANT	BQUANT
00	(5xQUANT)/4
01	(6xQUANT)/4
10	(7xQUANT)/4
11	(8xQUANT)/4

5.1.24 Extra Insertion Information (PEI) (1 bit)

A bit which when set to "1" signals the presence of the following optional data field.

5.1.25 Supplemental Enhancement Information (PSUPP) (0/8/16 ... bits)

If PEI is set to "1", then 9 bits follow consisting of 8 bits of data (PSUPP) and then another PEI bit to indicate if a further 9 bits follow and so on. Encoders shall use PSUPP as specified in Annex L. Decoders which do not support the extended capabilities described in Annex L shall be designed to discard PSUPP if PEI is set to 1. This enables backward compatibility for the extended capabilities of Annex L so that a bitstream which makes use of the extended capabilities can also be used without alteration by decoders which do not support those capabilities.

5.1.26 Stuffing (ESTUF) (Variable length)

A codeword of variable length consisting of less than 8 zero-bits. Encoders may insert this codeword directly before an EOS codeword. Encoders shall insert this codeword as necessary to attain mandatory byte alignment directly before an EOSBS codeword. If ESTUF is present, the last bit of ESTUF shall be the last (least significant) bit of a byte, so that the start of the EOS or EOSBS codeword is byte aligned. Decoders shall be designed to discard ESTUF. See Annex C for a description of EOSBS and its use.

5.1.27 End Of Sequence (EOS) (22 bits)

A codeword of 22 bits. Its value is 0000 0000 0000 0000 1 11111. It is up to the encoder to insert this codeword or not. EOS may be byte aligned. This can be achieved by inserting ESTUF before the start code such that the first bit of the start code is the first (most significant) bit of a byte. EOS shall not be repeated unless at least one picture start code appears between each pair of EOS codes.

5.1.28 Stuffing (PSTUF) (Variable length)

A codeword of variable length consisting of less than 8 zero-bits. Encoders shall insert this codeword for byte alignment of the next PSC. The last bit of PSTUF shall be the last (least significant) bit of a byte, so that the video bitstream including PSTUF is a multiple of 8 bits from the first bit in the H.263 bitstream. Decoders shall be designed to discard PSTUF.

If for some reason the encoder stops encoding pictures for a certain time-period and resumes encoding later, PSTUF shall be transmitted before the encoder stops, to prevent that the last up to 7 bits of the previous picture are not sent until the coder resumes coding.

5.2 Group of Blocks Layer

Data for each Group of Blocks (GOB) consists of a GOB header followed by data for macroblocks. The structure is shown in Figure 9. Each GOB contains one or more rows of macroblocks. For the first GOB in each picture (with number 0), no GOB header shall be transmitted. For all other GOBs, the GOB header may be empty, depending on the encoder strategy. A decoder can signal the remote encoder to transmit only non-empty GOB headers by external means, for example Recommendation H.245. GSTUF may be present when GBSC is present. GN, GFID and GQUANT are present when GBSC is present. GSBI is present when CPM is "1" in the Picture header.

FIGURE 9/H.263

Structure of GOB layer

5.2.1 Stuffing (GSTUF) (Variable length)

A codeword of variable length consisting of less than 8 zero-bits. Encoders may insert this codeword directly before a GBSC codeword. If GSTUF is present, the last bit of GSTUF shall be the last (least significant) bit of a byte, so that the start of the GBSC codeword is byte aligned. Decoders shall be designed to discard GSTUF.

5.2.2 Group of Block Start Code (GBSC) (17 bits)

A word of 17 bits. Its value is 0000 0000 0000 0000 1. GOB start codes may be byte aligned. This can be achieved by inserting GSTUF before the start code such that the first bit of the start code is the first (most significant) bit of a byte.

5.2.3 Group Number (GN) (5 bits)

A fixed length codeword of 5 bits. The bits are the binary representation of the number of the Group of Blocks. For the GOB with number 0, the GOB header including GSTUF, GBSC, GN, GSBI, GFID and GQUANT is empty; as group number 0 is used in the PSC. Group numbers 1 through 17 are used in GOB headers of the standard picture formats. Group numbers 1 through 24 are used in GOB headers of custom picture formats. Group numbers 16 through 28 are emulated in the slice header (see Annex K) when CPM='0', and Group Numbers 25-27 and 29 are emulated in the slice header (see Annex K) when CPM='1'. Group number 31 is used in the EOS code, and group number 30 is used in the EOSBS code.

5.2.4 GOB Sub-Bitstream Indicator (GSBI) (2 bits)

A fixed length codeword of 2 bits that is only present if CPM is "1" in the picture header. The codewords are the natural binary representation of the sub-bitstream number for the GOB header and all following information until the next Picture or GOB start code (see also Annex C).

5.2.5 GOB Frame ID (GFID) (2 bits)

A fixed length codeword of 2 bits. GFID shall have the same value in every GOB (or slice) header of a given picture. Moreover, if PTYPE as indicated in a picture header is the same as for the previous transmitted picture, GFID shall have the same value as in that previous picture, provided PLUSPTYPE is not present. However, if PTYPE in a certain picture header differs from the PTYPE in the previous transmitted picture header, the value for GFID in that picture shall differ from the value in the previous picture.

If PLUSPTYPE is present, the value of GFID shall be the same as that for the previous picture (in the same layer) if the PTYPE, and PLUSPTYPE, and all of the present fields among CPFMT, EPAR, CPCFC, SSS, ELNUM, RLNUM, UUI, RPSMF, and RPRP remain in effect as for the previous picture; otherwise, GFID shall be different than that for the previous picture.

5.2.6 Quantizer Information (GQUANT) (5 bits)

A fixed length codeword of 5 bits which indicates the quantizer QUANT to be used for the remaining part of the picture until updated by any subsequent GQUANT or DQUANT. The codewords are the natural binary representations of the values of QUANT which, being half the step sizes, range from 1 to 31.

5.3 Macroblock Layer

Data for each macroblock consists of a macroblock header followed by data for blocks. The structure is shown in Figure 10. COD is only present in pictures that are not of type 'INTRA', for each macroblock in these pictures. MCBPC is present when indicated by COD or when the picture is of type 'INTRA'. MODB is present for MB-type 0-4 if PTYPE indicates "PB-frame". CBPY, DQUANT, MVD and MVD_{2-4} are present when indicated by MCBPC. CBPB and MVDB are only present if indicated by MODB. Block Data is present when indicated by MCBPC and CBPY. MVD_{2-4} are only present when in Advanced Prediction mode (refer to Annex F) or Deblocking Filter mode (refer to Annex J). MODB, CBPB and MVDB are only present in PB-frames mode (refer to Annex G). For coding of the symbols in the Syntax-based Arithmetic Coding mode, refer to Annex E. For coding of the macroblock layer in B, EI, and EP pictures, see Annex O.

COD	MCBPC	MODB	CBPB	CBPY	DQUANT	MVD	MVD_2	MVD_3	MVD_4	MVDB	Block Data

FIGURE 10/H.263

Structure of macroblock layer

5.3.1 Coded macroblock indication (COD) (1 bit)

A bit which when set to "0" signals that the macroblock is coded. If set to "1", no further information is transmitted for this macroblock; in that case the decoder shall treat the macroblock as an INTER macroblock with motion vector for the whole block equal to zero and with no coefficient data. COD is only present in pictures that are not of type 'INTRA', for each macroblock in these pictures.

NOTE – In Advanced Prediction mode (see Annex F), overlapped block motion compensation is also performed if COD is set to "1"; and in Deblocking Filter mode (see Annex J), the deblocking filter is also performed if COD is set to "1".

5.3.2 Macroblock type & Coded block pattern for chrominance (MCBPC) (Variable length)

MCBPC is a variable length codeword giving information about the macroblock type and the coded block pattern for chrominance. The codewords for MCBPC are given in Tables 7 and 8. MCBPC is always included in coded macroblocks.

An extra codeword is available in the tables for bit stuffing. This codeword should be discarded by decoders. If an Improved PB-frame is indicated by MPPTYPE Bits 1-3 and Custom Source Format is indicated in OPPTYPE Bits 1-3, then MBA shall not indicate Stuffing before the first macroblock of the picture (in order to prevent start code emulation).

The macroblock type gives information about the macroblock and which data elements are present. Macroblock types and included elements are listed in Tables 9 and 10. Macroblock type 5 (indices 21-24 in Table 8) shall not be present unless an extended PTYPE (PLUSPTYPE) is present in the picture header and either the Advanced Prediction mode (see Annex F) or the Deblocking Filter mode (see Annex J) is in use, and shall not be present for the first macroblock of a picture. Also, encoders shall not allow an MCBPC code for macroblock type 5 to immediately follow seven consecutive zeros in the bitstream (as can be caused by particular INTRADC codes followed by COD=0), in order to prevent start code emulation. Codes for macroblock type 5 can be preceded by stuffing when necessary to fulfill this requirement (for macroblocks other than the first of a picture).

TABLE 7/H.263

VLC table for MCBPC (for I-pictures)

Index	MB type	CBPC (56)	Number of bits	Code
0	3	00	1	1
1	3	01	3	001
2	3	10	3	010
3	3	11	3	011
4	4	00	4	0001
5	4	01	6	0000 01
6	4	10	6	0000 10
7	4	11	6	0000 11
8	Stuffing	--	9	0000 0000 1

The coded block pattern for chrominance signifies C_B and/or C_R blocks when at least one non-INTRADC transform coefficient is transmitted (INTRADC is the dc-coefficient for INTRA blocks, see 5.4.1), unless the optional Advanced INTRA Coding mode is in use. $CBPC_N$ = 1 if any non-INTRADC coefficient is present for block N, else 0, for $CBPC_5$ and $CBPC_6$ in the coded block pattern. If Advanced INTRA Coding is in use, the usage is similar, but the INTRADC coefficient is indicated in the same way as the other coefficients (see Annex I). Block numbering is given in Figure 5. When MCBPC = Stuffing, the remaining part of the macroblock layer is skipped. In this case, the preceding COD = 0 is not related to any coded or not-coded macroblock and therefore the macroblock number is not incremented. For P-pictures, multiple stuffings are accomplished by multiple sets of COD=0 and MCBPC=Stuffing. See Tables 7 and 8.

TABLE 8/H.263

VLC table for MCBPC (for P-pictures)

Index	MB type	CBPC (56)	Number of bits	Code
0	0	00	1	1
1	0	01	4	0011
2	0	10	4	0010
3	0	11	6	0001 01
4	1	00	3	011
5	1	01	7	0000 111
6	1	10	7	0000 110
7	1	11	9	0000 0010 1
8	2	00	3	010
9	2	01	7	0000 101
10	2	10	7	0000 100
11	2	11	8	0000 0101
12	3	00	5	0001 1
13	3	01	8	0000 0100
14	3	10	8	0000 0011
15	3	11	7	0000 011
16	4	00	6	0001 00
17	4	01	9	0000 0010 0
18	4	10	9	0000 0001 1
19	4	11	9	0000 0001 0
20	Stuffing	--	9	0000 0000 1
21	5	00	11	0000 0000 010
22	5	01	13	0000 0000 0110 0
23	5	10	13	0000 0000 0111 0
24	5	11	13	0000 0000 0111 1

TABLE 9/H.263

Macroblock types and included data elements for normal pictures

Picture type	MB type	Name	COD	MCBPC	CBPY	DQUANT	MVD	MVD_{2-4}
INTER	not coded	-	X					
INTER	0	INTER	X	X	X		X	
INTER	1	INTER+Q	X	X	X	X	X	
INTER	2	INTER4V	X	X	X		X	X
INTER	3	INTRA	X	X	X			
INTER	4	INTRA+Q	X	X	X	X		
INTER	5	INTER4V+Q	X	X	X	X	X	X
INTER	stuffing	-	X	X				
INTRA	3	INTRA		X	X			
INTRA	4	INTRA+Q		X	X	X		
INTRA	stuffing	-		X				

Note: "x" means that the item is present in the macroblock

TABLE 10/H.263

Macroblock types and included data elements for PB-frames

Picture type	MB type	Name	COD	MCBPC	MODB	CBPY	CBPB	DQUANT	MVD	MVDB	MVD_{2-4}
INTER	not coded	-	X								
INTER	0	INTER	X	X	X	X	(X)		X	(X)	
INTER	1	INTER+Q	X	X	X	X	(X)	X	X	(X)	
INTER	2	INTER4V	X	X	X	X	(X)		X	(X)	X
INTER	3	INTRA	X	X	X	X	(X)		X	(X)	
INTER	4	INTRA+Q	X	X	X	X	(X)	X	X	(X)	
INTER	5	INTER4V+Q	X	X	X	X	(X)	X	X	(X)	X
INTER	stuffing	-	X	X							

Note 1: "x" means that the item is present in the macroblock

Note 2: CBPB and MVDB are only present if indicated by MODB

Note 3: B-blocks are always coded in INTER mode, even if the macroblock type of the PB-macroblock indicates INTRA

5.3.3 Macroblock mode for B-blocks (MODB) (Variable length)

MODB is present for MB-type 0-4 if PTYPE indicates 'PB-frame' and is a variable length codeword indicating whether CBPB is present (indicates that B-coefficients are transmitted for this macroblock) and/or MVDB is present. In Table 11 the codewords for MODB are defined. MODB is coded differently for Improved PB frames, as specified in Annex M.

TABLE 11/H.263

VLC table for MODB

Index	CBPB	MVDB	Number of bits	Code
0			1	0
1		X	2	10
2	X	X	2	11

Note: "X" means that the item is present in the macroblock

5.3.4 Coded block pattern for B-blocks (CBPB) (6 bits)

CBPB is only present in PB-frames mode if indicated by MODB. $CBPB_N$ = 1 if any coefficient is present for B-block N, else 0, for each bit $CBPB_N$ in the coded block pattern. Block numbering is given in Figure 5, the utmost left bit of CBPB corresponding with block number 1.

5.3.5 Coded block pattern for luminance (CBPY) (Variable length)

Variable length codeword giving a pattern number signifying those Y blocks in the macroblock for which at least one non-INTRADC transform coefficient is transmitted (INTRADC is the dc-coefficient for INTRA blocks, see section 5.4.1), unless the Advanced INTRA Coding mode is in use. If Advanced INTRA Coding is in use, INTRADC is indicated in the same manner as the other coefficients (see Annex I).

$CBPY_N$ = 1 if any non-INTRADC coefficient is present for block N, else 0, for each bit $CBPY_N$ in the coded block pattern. Block numbering is given in Figure 5, the utmost left bit of CBPY corresponding with block number 1. For a certain pattern $CBPY_N$, different codewords are used for INTER and INTRA macroblocks as defined in Table 13.

5.3.6 Quantizer Information (DQUANT) (2 bits/Variable Length)

If the Modified Quantization mode is not in use, DQUANT is a two bit code to define a change in QUANT. In Table 12 the differential values for the different codewords are given. QUANT ranges from 1 to 31; if the value for QUANT after adding the differential value is less than 1 or greater than 31, it is clipped to 1 and 31 respectively. If the Modified Quantization mode is in use, DQUANT is a variable length codeword as specified in Annex T.

TABLE 12/H.263

DQUANT codes and differential values for QUANT

Index	Differential value	DQUANT
0	-1	00
1	-2	01
2	1	10
3	2	11

TABLE 13/H.263

VLC table for CBPY

Index	CBPY(INTRA) (12 34)	CBPY(INTER) (12 34)	Number of bits	Code
0	00 00	11 11	4	0011
1	00 01	11 10	5	0010 1
2	00 10	11 01	5	0010 0
3	00 11	11 00	4	1001
4	01 00	10 11	5	0001 1
5	01 01	10 10	4	0111
6	01 10	10 01	6	0000 10
7	01 11	10 00	4	1011
8	10 00	01 11	5	0001 0
9	10 01	01 10	6	0000 11
10	10 10	01 01	4	0101
11	10 11	01 00	4	1010
12	11 00	00 11	4	0100
13	11 01	00 10	4	1000
14	11 10	00 01	4	0110
15	11 11	00 00	2	11

5.3.7 Motion Vector Data (MVD) (Variable length)

MVD is included for all INTER macroblocks (in PB-frames mode also for INTRA macroblocks) and consists of a variable length codeword for the horizontal component followed by a variable length codeword for the vertical component. Variable length codes are given in Table 14. If the Unrestricted Motion Vector mode is used and PLUSPTYPE is present, motion vectors are coded using Table D.3 instead of Table 14 (see Annex D).

TABLE 14/H.263

VLC table for MVD

Index	Vector differences		Bit number	Codes
0	-16	16	13	0000 0000 0010 1
1	-15.5	16.5	13	0000 0000 0011 1
2	-15	17	12	0000 0000 0101
3	-14.5	17.5	12	0000 0000 0111
4	-14	18	12	0000 0000 1001
5	-13.5	18.5	12	0000 0000 1011
6	-13	19	12	0000 0000 1101
7	-12.5	19.5	12	0000 0000 1111
8	-12	20	11	0000 0001 001
9	-11.5	20.5	11	0000 0001 011
10	-11	21	11	0000 0001 101
11	-10.5	21.5	11	0000 0001 111
12	-10	22	11	0000 0010 001
13	-9.5	22.5	11	0000 0010 011
14	-9	23	11	0000 0010 101
15	-8.5	23.5	11	0000 0010 111
16	-8	24	11	0000 0011 001
17	-7.5	24.5	11	0000 0011 011
18	-7	25	11	0000 0011 101
19	-6.5	25.5	11	0000 0011 111
20	-6	26	11	0000 0100 001
21	-5.5	26.5	11	0000 0100 011
22	-5	27	10	0000 0100 11
23	-4.5	27.5	10	0000 0101 01
24	-4	28	10	0000 0101 11
25	-3.5	28.5	8	0000 0111
26	-3	29	8	0000 1001
27	-2.5	29.5	8	0000 1011
28	-2	30	7	0000 111
29	-1.5	30.5	5	0001 1
30	-1	31	4	0011
31	-0.5	31.5	3	011
32	0		1	1
33	0.5	-31.5	3	010
34	1	-31	4	0010
35	1.5	-30.5	5	0001 0
36	2	-30	7	0000 110
37	2.5	-29.5	8	0000 1010
38	3	-29	8	0000 1000
39	3.5	-28.5	8	0000 0110
40	4	-28	10	0000 0101 10
41	4.5	-27.5	10	0000 0101 00
42	5	-27	10	0000 0100 10
43	5.5	-26.5	11	0000 0100 010
44	6	-26	11	0000 0100 000
45	6.5	-25.5	11	0000 0011 110
46	7	-25	11	0000 0011 100
47	7.5	-24.5	11	0000 0011 010
48	8	-24	11	0000 0011 000
49	8.5	-23.5	11	0000 0010 110
50	9	-23	11	0000 0010 100
51	9.5	-22.5	11	0000 0010 010
52	10	-22	11	0000 0010 000
53	10.5	-21.5	11	0000 0001 110
54	11	-21	11	0000 0001 100
55	11.5	-20.5	11	0000 0001 010
56	12	-20	11	0000 0001 000
57	12.5	-19.5	12	0000 0000 1110
58	13	-19	12	0000 0000 1100
59	13.5	-18.5	12	0000 0000 1010
60	14	-18	12	0000 0000 1000
61	14.5	-17.5	12	0000 0000 0110
62	15	-17	12	0000 0000 0100
63	15.5	-16.5	13	0000 0000 0011 0

5.3.8 Motion Vector Data (MVD_{2-4}) (Variable length)

The three codewords MVD_{2-4} are included if indicated by PTYPE and by MCBPC, and consist each of a variable length codeword for the horizontal component followed by a variable length codeword for the vertical component of each vector. Variable length codes are given in Table 14. MVD_{2-4} are only present when in Advanced Prediction mode (see Annex F) or Deblocking Filter mode (see Annex J).

5.3.9 Motion vector data for B-macroblock (MVDB) (Variable length)

MVDB is only present in PB-frames or Improved PB-frames mode if indicated by MODB, and consists of a variable length codeword for the horizontal component followed by a variable length codeword for the vertical component of each vector. Variable length codes are given in Table 14. For the use of MVDB, refer to Annex G and Annex M.

5.4 Block Layer

If not in PB-frames mode, a macroblock comprises four luminance blocks and one of each of the two colour difference blocks (see Figure 5). The structure of the block layer is shown in Figure 11. INTRADC is present for every block of the macroblock if MCBPC indicates macroblock type 3 or 4 (see Tables 7 and 8). TCOEF is present if indicated by MCBPC or CBPY.

In PB-frames mode, a macroblock comprises twelve blocks. First the data for the six P-blocks is transmitted as in the default H.263 mode, then the data for the six B-blocks. INTRADC is present for every P-block of the macroblock if MCBPC indicates macroblock type 3 or 4 (see Tables 7 and 8). INTRADC is not present for B-blocks. TCOEF is present for P-blocks if indicated by MCBPC or CBPY; TCOEF is present for B-blocks if indicated by CBPB.

For coding of the symbols in the Syntax-based Arithmetic Coding mode, refer to Annex E.

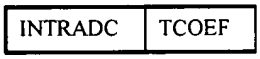


FIGURE 11/H.263

Structure of block layer

5.4.1 DC coefficient for INTRA blocks (INTRADC) (8 bits)

A codeword of 8 bits. The code 0000 0000 is not used. The code 1000 0000 is not used, the reconstruction level of 1024 being coded as 1111 1111 (see Table 15).

TABLE 15/H.263

Reconstruction levels for INTRA-mode DC coefficient

Index	FLC	Reconstruction level into inverse transform
0	0000 0001 (1)	8
1	0000 0010 (2)	16
2	0000 0011 (3)	24
.	. .	.
.	. .	.
126	0111 1111 (127)	1016
127	1111 1111 (255)	1024
128	1000 0001 (129)	1032
.	. .	.
.	. .	.
252	1111 1101 (253)	2024
253	1111 1110 (254)	2032

5.4.2 Transform coefficient (TCOEF) (Variable length)

The most commonly occurring EVENTs are coded with the variable length codes given in Table 16. The last bit "s" denotes the sign of the level, "0" for positive and "1" for negative.

An EVENT is a combination of a last non-zero coefficient indication (LAST; "0": there are more non-zero coefficients in this block, "1": this is the last non-zero coefficient in this block), the number of successive zeros preceding the coded coefficient (RUN), and the non-zero value of the coded coefficient (LEVEL).

The remaining combinations of (LAST, RUN, LEVEL) are coded with a 22 bit word consisting of 7 bits ESCAPE, 1 bit LAST, 6 bits RUN and 8 bits LEVEL. Use of this 22-bit word for encoding the combinations listed in Table 16 is not prohibited. For the 8-bit word for LEVEL, the code 0000 0000 is forbidden, and the code 1000 0000 is forbidden unless the Modified Quantization mode is in use (see Annex T). The codes for RUN and for LEVEL are given in Table 17.

TABLE 16/H.263

VLC table for TCOEF

INDEX	LAST	RUN	|LEVEL|	BITS	VLC CODE
0	0	0	1	3	10s
1	0	0	2	5	1111 s
2	0	0	3	7	0101 01s
3	0	0	4	8	0010 111s
4	0	0	5	9	0001 1111 s
5	0	0	6	10	0001 0010 1s
6	0	0	7	10	0001 0010 0s
7	0	0	8	11	0000 1000 01s
8	0	0	9	11	0000 1000 00s
9	0	0	10	12	0000 0000 111s
10	0	0	11	12	0000 0000 110s
11	0	0	12	12	0000 0100 000s
12	0	1	1	4	110s
13	0	1	2	7	0101 00s
14	0	1	3	9	0001 1110 s
15	0	1	4	11	0000 0011 11s
16	0	1	5	12	0000 0100 001s
17	0	1	6	13	0000 0101 0000s
18	0	2	1	5	1110 s
19	0	2	2	9	0001 1101 s
20	0	2	3	11	0000 0011 10s
21	0	2	4	13	0000 0101 0001s
22	0	3	1	6	0110 1s
23	0	3	2	10	0001 0001 1s
24	0	3	3	11	0000 0011 01s
25	0	4	1	6	0110 0s
26	0	4	2	10	0001 0001 0s
27	0	4	3	13	0000 0101 0010s
28	0	5	1	6	0101 1s
29	0	5	2	11	0000 0011 00s
30	0	5	3	13	0000 0101 0011s
31	0	6	1	7	0100 11s
32	0	6	2	11	0000 0010 11s

INDEX	LAST	RUN	|LEVEL|	BITS	VLC CODE
58	1	0	1	5	0111 s
59	1	0	2	10	0000 1100 1s
60	1	0	3	12	0000 0000 101s
61	1	1	1	7	0011 11s
62	1	1	2	12	0000 0000 100s
63	1	2	1	7	0011 10s
64	1	3	1	7	0011 01s
65	1	4	1	7	0011 00s
66	1	5	1	8	0010 011s
67	1	6	1	8	0010 010s
68	1	7	1	8	0010 001s
69	1	8	1	8	0010 000s
70	1	9	1	9	0001 1010 s
71	1	10	1	9	0001 1001 s
72	1	11	1	9	0001 1000 s
73	1	12	1	9	0001 0111 s
74	1	13	1	9	0001 0110 s
75	1	14	1	9	0001 0101 s
76	1	15	1	9	0001 0100 s
77	1	16	1	9	0001 0011 s
78	1	17	1	10	0000 1100 0s
79	1	18	1	10	0000 1011 1s
80	1	19	1	10	0000 1011 0s
81	1	20	1	10	0000 1010 1s
82	1	21	1	10	0000 1010 0s
83	1	22	1	10	0000 1001 1s
84	1	23	1	10	0000 1001 0s
85	1	24	1	10	0000 1000 1s
86	1	25	1	11	0000 0001 11s
87	1	26	1	11	0000 0001 10s
88	1	27	1	11	0000 0001 01s
89	1	28	1	11	0000 0001 00s
90	1	29	1	12	0000 0100 100s

TABLE 16/H.263

VLC table for TCOEF *(concluded)*

INDEX	LAST	RUN	|LEVEL|	BITS	VLC CODE
33	0	6	3	13	0000 0101 0100s
34	0	7	1	7	0100 10s
35	0	7	2	11	0000 0010 10s
36	0	8	1	7	0100 01s
37	0	8	2	11	0000 0010 01s
38	0	9	1	7	0100 00s
39	0	9	2	11	0000 0010 00s
40	0	10	1	8	0010 110s
41	0	10	2	13	0000 0101 0101s
42	0	11	1	8	0010 101s
43	0	12	1	8	0010 100s
44	0	13	1	9	0001 1100 s
45	0	14	1	9	0001 1011 s
46	0	15	1	10	0001 0000 1s
47	0	16	1	10	0001 0000 0s
48	0	17	1	10	0000 1111 1s
49	0	18	1	10	0000 1111 0s
50	0	19	1	10	0000 1110 1s
51	0	20	1	10	0000 1110 0s
52	0	21	1	10	0000 1101 1s
53	0	22	1	10	0000 1101 0s
54	0	23	1	12	0000 0100 010s
55	0	24	1	12	0000 0100 011s
56	0	25	1	13	0000 0101 0110s
57	0	26	1	13	0000 0101 0111s

INDEX	LAST	RUN	|LEVEL|	BITS	VLC CODE
91	1	30	1	12	0000 0100 101s
92	1	31	1	12	0000 0100 110s
93	1	32	1	12	0000 0100 111s
94	1	33	1	13	0000 0101 1000s
95	1	34	1	13	0000 0101 1001s
96	1	35	1	13	0000 0101 1010s
97	1	36	1	13	0000 0101 1011s
98	1	37	1	13	0000 0101 1100s
99	1	38	1	13	0000 0101 1101s
100	1	39	1	13	0000 0101 1110s
101	1	40	1	13	0000 0101 1111s
102	ESCAPE			7	0000 011

TABLE 17/H.263

FLC table for RUNS and LEVELS

Index	Run	Code
0	0	000 000
1	1	000 001
2	2	000 010
.	.	.
.	.	.
63	63	111 111

Index	Level	Code
-	-128	see text
0	-127	1000 0001
.	.	.
125	-2	1111 1110
126	-1	1111 1111
-	0	FORBIDDEN
127	1	0000 0001
128	2	0000 0010
.	.	.
253	127	0111 1111

6 Decoding process

6.1 Motion compensation

In this clause, the motion compensation for the default H.263 prediction mode is described. For a description of motion compensation in the Unrestricted Motion Vector mode, refer to Annex D. For a description of motion compensation in the Advanced Prediction mode, refer to Annex F. For a description of motion compensation in the Reduced-Resolution Update mode refer to Annex Q.

6.1.1 Differential motion vectors

The macroblock vector is obtained by adding predictors to the vector differences indicated by MVD (see Table 14 and Table D.3). For differential coding with four vectors per macroblock refer to Annex F. In case of one vector per macroblock, the candidate predictors for the differential coding are taken from three surrounding macroblocks as indicated in Figure 12. The predictors are calculated separately for the horizontal and vertical components.

In the special cases at the borders of the current GOB, slice, or picture, the following decision rules are applied in increasing order:

1) When the corresponding macroblock was coded in INTRA mode (if not in PB-frames mode with bidirectional prediction) or was not coded (COD = 1), the candidate predictor is set to zero.

2) The candidate predictor MV1 is set to zero if the corresponding macroblock is outside the picture or the slice (at the left side).

3) Then, the candidate predictors MV2 and MV3 are set to MV1 if the corresponding macroblocks are outside the picture (at the top) or outside the GOB (at the top) if the GOB header of the current GOB is non-empty; or outside the slice when in Slice Structured mode,

4) Then, the candidate predictor MV3 is set to zero if the corresponding macroblock is outside the picture (at the right side).

For each component, the predictor is the median value of the three candidate predictors for this component.

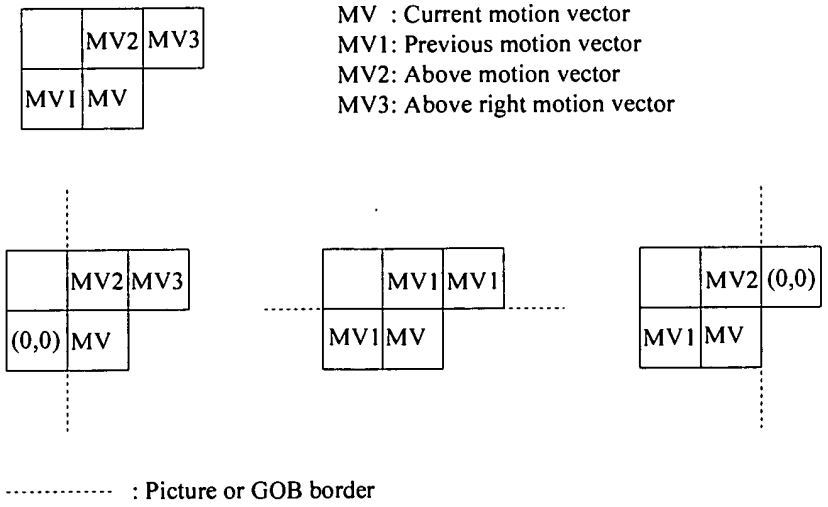


FIGURE 12/H.263

Motion vector prediction

Advantage is taken of the fact that the range of motion vector component values is constrained. Each VLC word for MVD represents a pair of difference values. Only one of the pair will yield a macroblock vector component falling within the permitted range [–16, 15.5]. A positive value of the horizontal or vertical component of the motion vector signifies that the prediction is formed from pixels in the previous picture which are spatially to the right or

below the pixels being predicted. If the unrestricted motion vector mode (see Annex D) is used, the decoding of motion vectors shall be performed as specified in section D.2.

The motion vector is used for all pixels in all four luminance blocks in the macroblock. Motion vectors for both chrominance blocks are derived by dividing the component values of the macroblock vector by two, due to the lower chrominance format. The component values of the resulting quarter pixel resolution vectors are modified towards the nearest half pixel position as indicated in Table 18.

TABLE 18/H.263

Modification of quarter pixel resolution chrominance vector components

quarter pixel position	0	1/4	1/2	3/4	1
resulting position	0	1/2	1/2	1/2	1

6.1.2 Interpolation for subpixel prediction

Half pixel values are found using bilinear interpolation as described in Figure 13. "/" indicates division by truncation.

The value of RCONTROL is equal to the value of the rounding type (RTYPE) bit (bit 6) in MPPTYPE (see section 5.1.4.3), when the Source Format field (bits 6-8) in PTYPE indicates "extended PTYPE". Otherwise RCONTROL has an implied value of 0. Regardless of the RTYPE bit, the value of RCONTROL is set to 0 for the B-part of Improved PB frames (see Annex M).

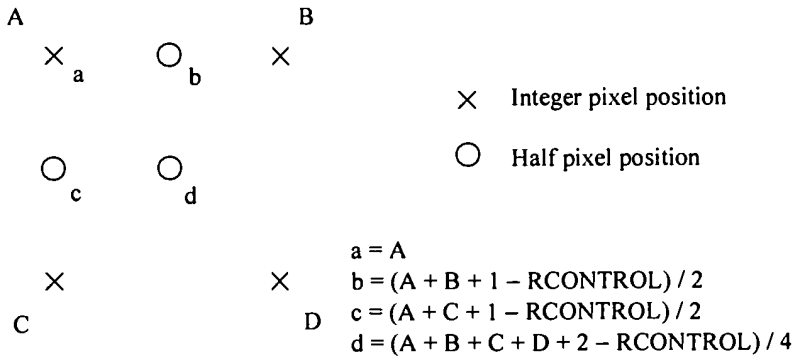


FIGURE 13/H.263

Halfpixel prediction by bilinear interpolation

6.2. Coefficients decoding

6.2.1. Inverse Quantization

The inverse quantization process is described in this section, except for when the optional Advanced INTRA Coding mode is in use (see Annex I). If LEVEL = "0", the reconstruction level REC = "0". The reconstruction level of INTRADC is given by Table 15. The reconstruction levels of all non-zero coefficients other than the INTRADC one are given by the following formulas:

$$|REC| = QUANT \cdot (2 \cdot |LEVEL| + 1) \qquad \text{if QUANT = "odd"}$$

$$|REC| = QUANT \cdot (2 \cdot |LEVEL| + 1) - 1 \qquad \text{if QUANT = "even"}$$

Note that this process disallows even valued numbers. This has been found to prevent accumulation of IDCT mismatch errors. After calculation of |REC|, the sign is added to obtain REC: REC = sign(LEVEL) · |REC|

Sign(LEVEL) is given by the last bit of the TCOEF code (see Table 16) or by Table 17.

6.2.2 Clipping of reconstruction levels

After inverse quantization, the reconstruction levels of all coefficients other than the INTRADC one are clipped to the range –2048 to 2047.

6.2.3 Zig-zag positioning

The quantized transform coefficients are placed into an 8 × 8 block according to the sequence given in Figure 14, unless the optional Advanced INTRA Coding mode is in use (see Annex I). Coefficient 1 is the dc-coefficient.

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

FIGURE 14/H.263

Zig-zag positioning of quantized transform coefficients

6.2.4 Inverse transform

After inverse quantization and zig-zag of coefficients, the resulting 8 × 8 blocks are processed by a separable two-dimensional inverse discrete cosine transform of size 8 by 8. The output from the inverse transform ranges from –256 to +255 after clipping to be represented with 9 bits. The transfer function of the inverse transform is given by:

$$f(x,y) = 1/4 \sum_{u=0}^{7} \sum_{v=0}^{7} C(u)C(v)F(u,v)\cos[\pi(2x+1)u/16]\cos[\pi(2y+1)v/16]$$

with $u, v, x, y = 0, 1, 2, \ldots, 7$

where x,y = spatial coordinates in the pixel domain,

u,v = coordinates in the transform domain,

$C(u) = 1/\sqrt{2}$ for $u = 0$, otherwise 1,

$C(v) = 1/\sqrt{2}$ for $v = 0$, otherwise 1.

NOTE – Within the block being transformed, $x = 0$ and $y = 0$ refer to the pixel nearest the left and top edges of the picture respectively.

The arithmetic procedures for computing the inverse transform are not defined, but should meet the error tolerance specified in Annex A.

6.3 Reconstruction of blocks

6.3.1 Summation

After motion compensation and coefficients decoding (inverse transform included), a reconstruction is formed for each luminance and chrominance block. For INTRA blocks, the reconstruction is equal to the result of the inverse

transformation. For INTER blocks, the reconstruction is formed by summing the prediction and the result of the inverse transformation. The summation is performed on a pixel basis. For the summation in the Reduced-Resolution Update mode refer to Annex Q.

6.3.2 Clipping

To prevent quantization distortion of transform coefficient amplitudes causing arithmetic overflow in the encoder and decoder loops, clipping functions are inserted. The clipper operates after the summation of prediction and reconstructed prediction error on resulting pixel values less than 0 or greater than 255, changing them to 0 and 255 respectively.

Annex A

Inverse transform accuracy specification

(This annex forms an integral part of this Recommendation)

A.1 Generate random integer pixel data values in the range –L to +H according to the random number generator given below ("C" version). Arrange into 8 by 8 blocks. Data set of 10 000 blocks should each be generated for (L = 256, H = 255), (L = H = 5) and (L = H = 300).

A.2 For each 8 by 8 block, perform a separable, orthonormal, matrix multiply, forward discrete cosine transform using at least 64-bit floating point accuracy.

$$F(u,v) = 1/4C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos[\pi(2x+1)u/16]\cos[\pi(2y+1)v/16]$$

with $u, v, x, y = 0, 1, 2, \ldots, 7$

x, y = spatial coordinates in the pixel domain;

u, v = coordinates in the transform domain;

$C(u) = 1/\sqrt{2}$ for $u = 0$, otherwise 1;

$C(v) = 1/\sqrt{2}$ for $v = 0$, otherwise 1.

A.3 For each block, round the 64 resulting transformed coefficients to the nearest integer values. Then clip them to the range –2048 to +2047. This is the 12-bit input data to the inverse transform.

A.4 For each 8 by 8 block of 12-bit data produced by A.3, perform a separable, orthonormal, matrix multiply, inverse discrete transform (IDCT) using at least 64-bit floating point accuracy. Round the resulting pixels to the nearest integer and clip to the range –256 to +255. These blocks of 8 × 8 pixels are the reference IDCT output data.

A.5 For each 8 by 8 block produced by A.3, apply the IDCT under test and clip the output to the range –256 to +255. These blocks of 8 × 8 pixels are the test IDCT output data.

A.6 For each of the 64 IDCT output pixels, and for each of the 10 000 block data sets generated above, measure the peak, mean and mean square error between the reference and the test data.

A.7
- For any pixel, the peak error should not exceed 1 in magnitude.
- For any pixel, the mean square error should not exceed 0.06.
- Overall, the mean square error should not exceed 0.02.
- For any pixel, the mean error should not exceed 0.015 in magnitude.
- Overall, the mean error should not exceed 0.0015 in magnitude.

A.8 All zeros in shall produce all zeros out.

A.9 Re-run the measurements using exactly the same data values of A.1, but change the sign on each pixel.

"C" program for random number generation

```
/* L and H shall be long, that is 32 bits */
long rand   (L,H)
long        L,H;
{
```

```
    static long randx = 1;                  /* long is 32 bits          */
    static double z = (double) 0x7fffffff;

    long        i,j;
    double      x;                          /* double is 64 bits        */

    randx = (randx * 1103515245) + 12345;
    i = randx & 0x7ffffffe;                 /* keep 30 bits             */
    x = ( (double)i ) /  z;                 /* range 0 to 0.99999 ...   */
    x *= (L+H+1);                           /* range 0 to < L+H+1       */
    j = x;                                  /* truncate to integer      */
    return(j – L);                          /* range –L to H            */
}
```

Annex B

Hypothetical Reference Decoder

(This annex forms an integral part of this Recommendation)

The Hypothetical Reference Decoder (HRD) is defined as follows:

B.1 The HRD and the encoder have the same clock frequency as well as the same picture clock frequency, and are operated synchronously.

B.2 The HRD receiving buffer size is (B + BPPmaxKb * 1024 bits) where (BPPmaxKb * 1024) is the maximum number of bits per picture that has been negotiated for use in the bitstream (see 3.6). The value of B is defined as follows:

$$B = 4 \cdot R_{max} / PCF$$

where PCF is the effective picture clock frequency, and R_{max} is the maximum video bit rate during the connection in bits per second. The effective picture clock frequency is the standard CIF picture clock frequency unless a custom PCF is specified in the CPCFC field of the picture header. This value for B is a minimum. An encoder may use a larger value for B, provided the larger number is first negotiated by external means, for example Recommendation H.245.

The value for R_{max} depends on the system configuration (for example GSTN or ISDN, single or multi-link) and may be equal to the maximum bit rate supported by the physical link. Negotiation of R_{max} is done by external means, for example Recommendation H.245.

B.3 The HRD is initially empty.

B.4 The HRD buffer is examined at picture clock intervals (1000 / PCF ms). If at least one complete coded picture is in the buffer then all the data for the earliest picture in bitstream order is instantaneously removed (e.g. at t_{n+1} in Figure B.1). Immediately after removing the above data the buffer occupancy must be less than B. This is a requirement on the coder output bitstream including coded picture data and MCBPC and STUF stuffing but not error correction framing bits, fill indicator (Fi), fill bits or error correction parity information described in Annex H.

For the purposes of this definition, unless the optional Temporal, SNR, and Spatial Scalability mode is in use, a complete coded picture is one normal I- or P-picture, or a PB-frame or Improved PB-frame.

When the Temporal, SNR, and Spatial Scalability mode is in use (see Annex O), each enhancement layer is given an additional HRD, for which a complete coded picture is one EI, EP, or B picture. The base layer buffer shall hold the bits of the picture header as they arrive, until a sufficient amount of the picture header has arrived to determine whether the picture is a base layer or enhancement layer picture, and the number of the enhancement layer. When it can be determined that the arriving picture belongs to an enhancement layer, all bits for that picture shall be instantly transferred to the appropriate enhancement layer HRD, and any later bits which arrive continue to be placed into the enhancement layer HRD until a sufficient amount of some new picture header has arrived to determine that the bistream should again be re-routed into another HRD buffer. The process of enhancement layer identification is instantaneous and asynchronous, independent of the picture clock interval checking times.

To meet this requirement the number of bits for the $(n+1)$th coded picture d_{n+1} must satisfy:

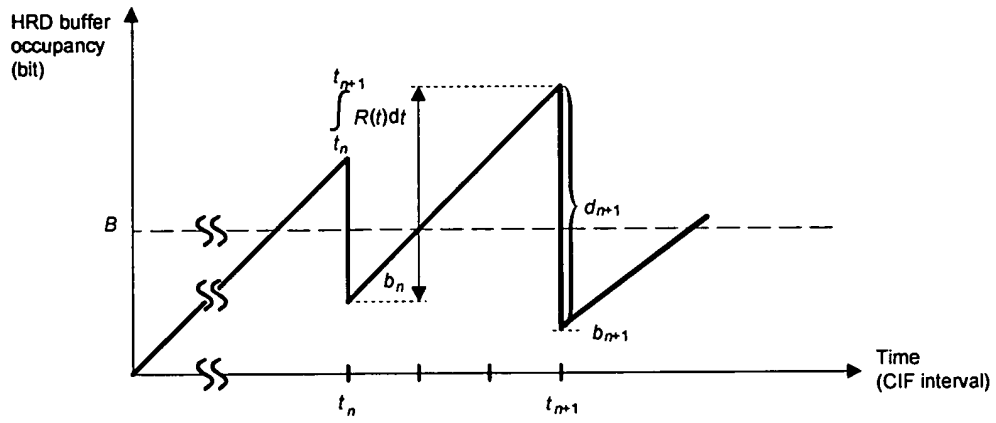
$$d_{n+1} \geq b_n + \int_{t_n}^{t_{n+1}} R(t)dt - B$$

where

b_n is the buffer occupancy just after time t_n;

t_n is the time the nth coded picture is removed from the HRD buffer;

$R(t)$ is the video bitrate at time t.



NOTE – Time $(t_{n+1} - t_n)$ is an integer number of CIF picture periods (1/29.97, 2/29.97, 3/29.97, ...).

FIGURE B.1/H.263

HRD buffer occupancy

Annex C

Considerations for Multipoint

(This annex forms an integral part of this Recommendation)

The following facilities are provided to support switched multipoint operation.

C.1 Freeze picture request

Causes the decoder to freeze its displayed picture until a freeze picture release signal is received or a time-out period of at least six seconds has expired. The transmission of this signal is by external means (for example Recommendation H.245). Note that a similar command may also be sent using supplemental enhancement information within the picture header of the video bitstream (see Annex L, section L.4).

C.2 Fast update request

Causes the encoder to encode its next picture in INTRA mode with coding parameters such as to avoid buffer overflow. The transmission method for this signal is by external means (for example Recommendation H.245).

C.3 Freeze picture release

A signal from an encoder which has responded to a fast update request and allows a decoder to exit from its freeze picture mode and display decoded pictures in the normal manner. This signal is transmitted by PTYPE (see 5.1.3) in the picture header of the first picture coded in response to the fast update request.

C.4 Continuous Presence Multipoint and Video Multiplexing (CPM)

(NOTE: Not used for H.324.)

In this Recommendation a negotiable Continuous Presence Multipoint and Video Multiplexing mode is provided in which up to four independent H.263 bitstreams can be multiplexed as independent "Sub-Bitstreams" in one new video bitstream with use of the PSBI, GSBI, SSBI, and ESBI fields. Capability exchange for this mode is done by external means (for example Recommendation H.242). Reference Picture Selection back-channel data for responding to independent Sub-Bitstreams is supported using the BCPM and BSBI fields.

When in CPM mode, the CPM field shall be set to "1" in each of the independent H.263 Sub-Bitstreams. Sub-Bitstreams are identified by a stream identifier number using the Sub-Bitstream indicators (SBIs) in the Picture and GOB or slice and EOSBS headers of each H.263 bitstream. The SBI indicates the number of the H.263 bitstream to which that header and all following information until the next Picture or GOB header or slice header in the composed video bitstream belongs.

Each Sub-Bitstream is considered as a normal H.263 bitstream and shall therefore comply to the capabilities that are exchanged by external means. The information for the different H.263 bitstreams is not transmitted in any special predefined order, an SBI can have any value independent from preceding SBIs and the picture rates for the different H.263 bitstreams may be different. The information in each individual bitstream is also completely independent from the information in the other bitstreams. For example, the GFID codewords in one Sub-Bitstream are not influenced by GFID or PTYPE codewords in other Sub-Bitstreams. Similarly, the mode state inference rules when using an extended picture type (PLUSPTYPE) in the picture header and all other aspects of the video bitstream operation shall operate independently and separately for each Sub-Bitstream.

As a matter of convention, the Sub-Bitstream with the lowest Sub-Bitstream identifier number (sent in SBI) is considered to have the highest priority in situations in which some conflict of resource requirements may necessitate a choice of priority (unless a different priority convention is established by external means).

In order to mark the end of each Sub-Bitstream of the CPM mode, a syntax is provided as shown in Figure C.1, provided the ability to send this added syntax is first negotiated by external means (while CPM operation was defined in Version 1 of this Recommendation, the end of Sub-Bitstream syntax was added in Version 2 and thus is not considered a part of Version 1 CPM operation). The end of Sub-Bitstream syntax (ESTUF+EOSBS+ESBI) marks the end of each Sub-Bitstream, rather than the end of the entire stream as done by an EOS code.

Note — No ability to negotiate CPM Sub-Bitstream operation as defined herein for Recommendation H.263 was adopted into any ITU-T Recommendations for terminals (such as Recommendation H.324) prior to the creation of Version 2 of this Recommendation. Thus any external negotiation of CPM operation adopted into a future H-series terminal Recommendation shall imply support for the end of Sub-Bitstream syntax unless otherwise specified in the H-series terminal Recommendation.

There are three parts to the end of Sub-Bitstream syntax. Following mandatory byte alignment using ESTUF, an EOSBS codeword of 23 bits is sent (corresponding to a GOB header with GN=30, which is otherwise unused in the syntax, followed by a single zero-valued bit which is reserved for future use). The EOSBS codeword is then followed by a two bit ESBI codeword indicating which Sub-Bitstream is affected. This pair of codewords signifies that the sending of data for the associated Sub-Bitstream has ended and that any subsequent data sent for the same Sub-Bitstream shall be entirely independent from that which came before the EOSBS. In particular, the next picture for the Sub-Bitstream after the EOSBS code shall not be an INTER picture or any other picture type which can use forward temporal prediction (an I or EI picture is allowed, but a P picture, PB-frame, Improved PB frame, B picture, or EP picture is not).

The syntax of EOSBS and ESBI is described in the following subsections. ESTUF is described in section 5.1.26.

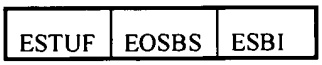


FIGURE C.1/H.263

Syntax diagram for End of Sub-Bitstream Indicators

C.4.1. End Of Sub-Bitstream code (EOSBS) (23 bits)

The EOSBS code is a codeword of 23 bits. Its value is 0000 0000 0000 0000 1 11110 0. It is up to the encoder whether to insert this codeword or not. EOSBS should not be sent unless at least one picture header has previously been sent for the same Sub-Bitstream indicated in the following ESBI field, and shall not be sent unless the capability to send EOSBS has been negotiated by external means. EOSBS codes shall be byte aligned. This is achieved by inserting ESTUF before the EOSBS start code such that the first bit of the EOSBS start code is the first (most significant) bit of a byte (see section 5.1.26).

The EOSBS code indicates that the sending of data for the indicated Sub-Bitstream has stopped and the Sub-Bitstream has been declared over, until re-started by issuance of another picture start code for that Sub-Bitstream. Subsequent pictures with the same Sub-Bitstream identifier number (ESBI) shall be completely independent of and shall not depend in any way on the pictures which were sent prior to the EOSBS code.

Control and other information associated with the video bitstream in general without specification of which Sub-Bitstreams these codes are meant to apply (such as a Freeze Picture Request or a Fast Update Request sent in H.242) should be presumed to apply to all of the *active* Sub-Bitstreams only. A Sub-Bitstream is considered active if at least one picture start code has been received for the Sub-Bitstream and the last data sent which was applicable to that Sub-Bitstream was not an EOS or EOSBS+ESBI.

C.4.2. Ending Sub-Bitstream Indicator (ESBI) (2 bits)

ESBI is a fixed length codeword of two bits which follows immediately after EOSBS. It indicates the Sub-Bitstream number of the ending Sub-Bitstream. Its value is the natural two bit binary representation of the Sub-Bitstream number.

Annex D

Unrestricted Motion Vector mode

(This annex forms an integral part of this Recommendation)

This annex describes the optional Unrestricted Motion Vector mode of H.263. The capability of this mode of H.263 is signalled by external means (for example Recommendation H.245). The use of this mode is indicated in PTYPE or PLUSPTYPE.

The range of motion vectors and the VLC table used for coding the motion vector differences for the Unrestricted Motion Vector mode depend on whether the PLUSPTYPE field of the picture header is present. When PLUSPTYPE is present, the range of motion vectors also depends on the picture size and the value of the UUI field in the picture header.

D.1 Motion vectors over picture boundaries

In the default prediction mode of H.263, motion vectors are restricted such that all pixels referenced by them are within the coded picture area (see 4.2.3). In the Unrestricted Motion Vector mode however this restriction is removed and therefore motion vectors *are* allowed to point outside the picture. When a pixel referenced by a motion vector is outside the coded picture area, an edge pixel is used instead. This edge pixel is found by limiting the motion vector to the last full pixel position inside the coded picture area. Limitation of the motion vector is performed on a pixel basis and separately for each component of the motion vector.

For example, if the Unrestricted Motion Vector mode is used for a QCIF picture, the referenced pixel value for the luminance component is given by the following formula:

$$Rumv(x, y) = R(x', y')$$

where:

x, y, x', y' = spatial coordinates in the pixel domain;

$Rumv(x, y)$ = pixel value of the referenced picture at (x, y) when in Unrestricted Motion Vector mode;

$R(x', y')$ = pixel value of the referenced picture at (x', y') when in Unrestricted Motion Vector mode;

$x' = 0$ if $x < 0$;

$= 175$ if $x > 175$;

$= x$ otherwise;

$y' = 0$ if $y < 0$;

$= 143$ if $y > 143$;

$= y$ otherwise;

and the coded picture area of R(x', y') is $0 <= x' <= 175$, $0 <= y' <= 143$. The given boundaries are integer pixel positions; however, (x', y') can also be a half pixel position within these boundaries.

D.1.1. Restrictions for motion vector values

If PLUSPTYPE is present in the picture header, the motion vector values are restricted such that no element of the 16×16 (or 8×8) region that is selected shall have a horizontal or vertical distance more than 15 pixels outside the coded picture area. Note that this is a smaller extrapolation range than when PLUSPTYPE is not present.

Note — When PLUSPTYPE is absent, the extrapolation range is a maximum of 31.5 pixels outside the coded picture area when the Unrestricted Motion Vector mode is used, and 16 pixels outside the coded picture area when the Advanced Prediction mode (see Annex F) is used without the Unrestricted Motion Vector mode.

D.2 Extension of the motion vector range

In the default prediction mode, the values for both horizontal and vertical components of the motion vectors are restricted to the range [–16, 15.5] (this is also valid for the forward and backward motion vector components for

B-pictures). In the Unrestricted Motion Vector mode however, the maximum range for vector components is extended.

If the PLUSPTYPE field is not present in the picture header, the motion vector range is extended to [–31.5, 31.5], with the restriction that only values that are within a range of [–16, 15.5] around the predictor for each motion vector component can be reached if the predictor is in the range [–15.5, 16]. If the predictor is outside [–15.5, 16], all values within the range [–31.5, 31.5] with the same sign as the predictor plus the zero value can be reached. So, if MV_C is the motion vector component and P_C is the predictor for it, then:

$$-31.5 \leq MV_C \leq 0 \qquad if -31.5 \leq P_C \leq -16$$

$$-16 + P_C \leq MV_C \leq 15.5 + P_C \qquad if -15.5 \leq P_C \leq 16$$

$$0 \leq MV_C \leq 31.5 \qquad if\ 16.5 \leq P_C \leq 31.5$$

In the Unrestricted Motion Vector mode, the interpretation of Table 14 for MVD, $MVD_{2\text{-}4}$ and MVDB is as follows:

- if the predictor for the motion vector component is in the range [–15.5, 16], only the first column of vector differences applies;
- if the predictor for the motion vector component is outside the range [–15.5, 16], the vector difference from Table 14 shall be used that results in a vector component inside the range [–31.5, 31.5] with the same sign as the predictor (including zero).

The predictor for MV_D and $MVD_{2\text{-}4}$ is defined as the median value of the vector components MV1, MV2 and MV3 as defined in 6.1.1 and F.2. For MVDB the predictor $P_C = (TR_B \times MV)/TR_D$ where MV represents a vector component for an 8 * 8 luminance block in a P-picture (see also G.4).

If PLUSPTYPE is present, the motion vector range does not depend on the motion vector prediction value. If the UUI field is set to "1" the motion vector range depends on the picture format. For standardized picture formats up to CIF the range is [-32,31.5], for those up to 4CIF the range is [-64,63.5], and for those up to 16CIF the range is [-128,127.5], and for even larger custom picture formats the range is [-256,255.5]. The horizontal and vertical motion vector ranges may be different for custom picture formats. The horizontal and vertical ranges are specified in Tables D.1 and D.2.

TABLE D.1/H.263

Horizontal Motion Vector Range when PLUSPTYPE present and UUI = 1

Picture width	Horizontal motion vector range
4, ..., 352	[-32, 31.5]
356, ..., 704	[-64, 63.5]
708, ..., 1408	[-128, 127.5]
1412,..., 2048	[-256, 255.5]

TABLE D.2/H.263

Vertical Motion Vector Range when PLUSPTYPE present and UUI = 1

Picture height	Vertical motion vector range
4, ..., 288	[-32, 31.5]
292, ..., 576	[-64, 63.5]
580, ..., 1152	[-128, 127.5]

In the Reduced-Resolution Update mode, the specified range applies to the pseudo motion vectors. This implies that the resulting actual motion vector range is enlarged to approximately double size (see also Annex Q).

If UUI is set to "01", the motion vectors are not limited except by their distance to the coded area border as explained in section D.1.1. The same limitation applies to the actual motion vectors (not just the pseudo motion vectors) in the Reduced-Resolution Update mode.

To encode the motion vectors when PLUSPTYPE is present, Table D.3 is used to encode the difference between the motion vector and the motion vector prediction. Every entry in Table D.3 has a single value (in contrast to Table 14). The motion vector range and the use of Table D.3 to encode motion vector data apply to all picture types when PLUSPTYPE is present.

Motion vectors differences are always coded as a pair of a horizontal and a vertical component. If a pair equals (0.5,0.5) six consecutive zeros are produced. To prevent start code emulation, this occurrence shall be followed by one bit set to "1". This corresponds to sending one additional zero motion vector component.

TABLE D.3/H.263

Motion Vector Table used when PLUSPTYPE present

Absolute value of vector difference in half-pixel units	number of bits	Codes
0	1	1
1	3	0s0
"x_0"+2 (2:3)	5	0$x_0$1s0
"x_1x_0"+4 (4:7)	7	0$x_1$1$x_0$1s0
"$x_2x_1x_0$"+8 (8:15)	9	0$x_2$1$x_1$1$x_0$1s0
"$x_3x_2x_1x_0$"+16 (16:31)	11	0$x_3$1$x_2$1$x_1$1$x_0$1s0
"$x_4x_3x_2x_1x_0$"+32 (32:63)	13	0$x_4$1$x_3$1$x_2$1$x_1$1$x_0$1s0
"$x_5x_4x_3x_2x_1x_0$"+64 (64:127)	15	0$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$1s0
"$x_6x_5x_4x_3x_2x_1x_0$"+128 (128:255)	17	0$x_6$1$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$1s0
"$x_7x_6x_5x_4x_3x_2x_1x_0$"+256 (256:511)	19	0$x_7$1$x_6$1$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$1s0
"$x_8x_7x_6x_5x_4x_3x_2x_1x_0$"+512 (512:1023)	21	0$x_8$1$x_7$1$x_6$1$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$1s0
"$x_9x_8x_7x_6x_5x_4x_3x_2x_1x_0$"+1024 (1024:2047)	23	0$x_9$1$x_8$1$x_7$1$x_6$1$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$1s0
"$x_{10}x_9x_8x_7x_6x_5x_4x_3x_2x_1x_0$"+2048 (2048:4095)	25	0x_{10}1$x_9$1$x_8$1$x_7$1$x_6$1$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$1s0

Table D.3 is a regularly constructed reversible table. Each row represents an interval of motion vector differences in half-pixel units. The bits "...x_1x_0" denote all bits following the leading "1" in the binary representation of the absolute value of the motion vector difference. The bit "s" denotes the sign of the motion vector difference, "0" for

positive and "1" for negative. The binary representation of the motion vector difference is interleaved with bits that indicate if the code continues or ends. For example, the motion vector difference −13 has sign s=1 and binary representation $1x_2x_1x_0 = 1101$. It is encoded as 0 $x_2$1 $x_1$1 $x_0$1 s0 = 0 11 11 01 10. The 0 in the second position of the last group of two bits indicates the end of the code.

Annex E

Syntax-based Arithmetic Coding mode

(This annex forms an integral part of this Recommendation)

E.1 Introduction

In the Variable Length Coding/Decoding (VLC/VLD) as described in clause 5, a symbol is VLC encoded using a specific table based on the syntax of the coder. This table typically stores lengths and values of the VLC code words. The symbol is mapped to an entry of the table in a table look-up operation, and then the binary code word specified by the entry is sent out normally to a buffer for transmitting to the receiver. In VLD decoding, the received bitstream is matched entry by entry in a specific table based on the syntax of the coder. This table must be the same as the one used in the encoder for encoding the current symbol. The matched entry in the table is then mapped back to the corresponding symbol which is the end result of the VLD decoder and is then used for recovering the video pictures. This VLC/VLD process implies that each symbol must be encoded into a fixed integral number of bits. Removing this restriction of fixed integral number of bits for symbols can lead to reductions of resulting bitrates, which can be achieved by arithmetic coding.

This annex describes the optional Syntax-based Arithmetic Coding (SAC) mode of H.263. In this mode, all the corresponding variable length coding/decoding operations of H.263 are replaced with arithmetic coding/decoding operations. The capability of this mode of H.263 is signalled by external means (for example Recommendation H.245). The use of this mode is indicated in PTYPE.

E.2 Specification of SAC encoder

In SAC mode, a symbol is encoded by using a specific array of integers (or a model) based on the syntax of the coder and by calling the following procedure which is specified in C language.

```
#define   q1    16384
#define   q2    32768
#define   q3    49152
#define   top   65535

static long      low, high, opposite_bits, length;

void   encode_a_symbol(int index, int cumul_freq[ ])

{
       length = high - low + 1;
       high = low - 1 + (length * cumul_freq[index]) / cumul_freq[0];
       low += (length * cumul_freq[index+1]) / cumul_freq[0];
       for ( ; ; ) {
          if (high < q2) {
             send out a bit "0" to PSC_FIFO;
             while (opposite_bits > 0) {
                send out a bit "1" to PSC_FIFO;
                opposite_bits--;
             }
          }
          else if (low >= q2) {
             send out a bit "1" to PSC_FIFO;
             while (opposite_bits > 0) {
                send out a bit "0" to PSC_FIFO;
                opposite_bits--;
             }
             low-= q2;
             high -= q2;
          }
```

```
            else if (low >= q1 && high < q3) {
                opposite_bits += 1;
                low -= q1;
                high -= q1;
            }
            else break;

            low *= 2;
            high = 2*high+1;
        }
}
```

The values of low, high and opposite_bits are initialized to 0, top and 0, respectively. PSC_FIFO is a FIFO for buffering the output bits from the arithmetic encoder. The model is specified through cumul_freq[ ], and the symbol is specified using its index in the model.

E.3 Specification of SAC decoder

In SAC decoder, a symbol is decoded by using a specific model based on the syntax and by calling the following procedure which is specified in C language.

```
static long      low, high, code_value, bit, length, index, cum;

int     decode_a_symbol(int cumul_freq[ ])

{
        length = high - low + 1;
        cum = (-1 + (code_value - low + 1) * cumul_freq[0]) / length;
        for (index = 1; cumul_freq[index] > cum; index++);
        high = low - 1 + (length * cumul_freq[index-1]) / cumul_freq[0];
        low += (length * cumul_freq[index]) / cumul_freq[0];
        for ( ; ; ) {
            if (high < q2);
            else if (low >= q2) {
                code_value -= q2;
                low -= q2;
                high -= q2;
            }
            else if (low >= q1 && high < q3) {
                code_value -= q1;
                low -= q1;
                high -= q1;
            }
            else break;

            low *= 2;
            high = 2 * high + 1;
            get bit from PSC_FIFO;
            code_value = 2 * code_value + bit;
        }
        return (index-1);
}
```

Again the model is specified through cumul_freq[ ]. The decoded symbol is returned through its index in the model. PSC_FIFO is a FIFO for buffering the incoming bitstream. The decoder is initialized to start decoding an arithmetic coded bitstream by calling the following procedure.

```
void    decoder_reset( )
{
        code_value = 0;
        low = 0;
        high = top;
        for (int i = 1;   i <= 16;   i++) {
            get bit from PSC_FIFO;
            code_value = 2 * code_value + bit;
        }
}
```

E.4 Syntax

As in VLC table mode of H.263, the syntax of the symbols is partitioned into four layers: Picture, Group of Blocks, Macroblock and Block. The syntax of the top three layers remain exactly the same. The syntax of the Block layer also remains quite similar, but is illustrated in Figure E.1.

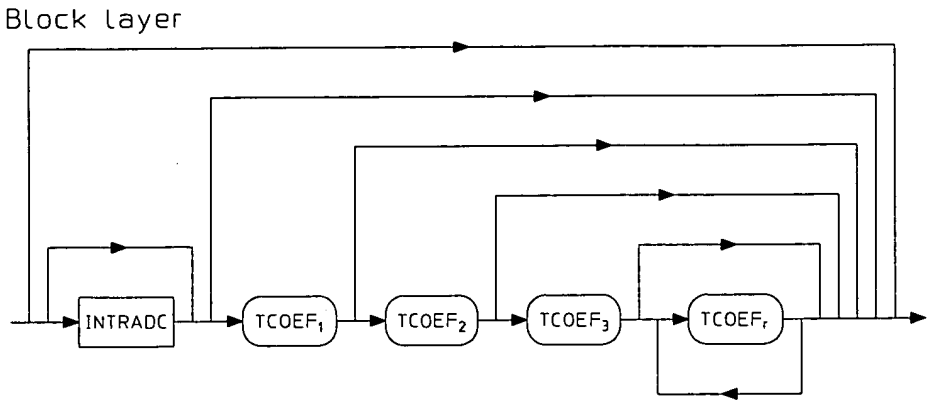


FIGURE E.1/H.263

Structure of SAC Block layer

In Figure E.1, TCOEF1, TCOEF2, TCOEF3 and TCOEFr are Last-Run-Level symbols as defined in 5.4.2, and are the possible 1st, 2nd, 3rd and rest of the symbols, respectively. TCOEF1, TCOEF2, TCOEF3 and TCOEFr are only present when one, two, three or more coefficients are present in the block layer, respectively.

E.5 PSC_FIFO

PSC_FIFO in encoder or decoder is a FIFO of size > 17 bits. In PSC_FIFO of encoder, illegal emulations of PSC and GBSC are located and are avoided by stuffing a "1" after each successive appearance of 14 "0"s (which are not part of PSC or GBSC). In PSC_FIFO of the decoder, the first "1" after each string of 14 "0"s is deleted; if instead a string of 14 "0"s is followed by a "0", it indicates that a legal PSC or GBSC is detected. The exact location of the PSC or GBSC is determined by the next "1" following the string of zeros.

E.6 Header layer symbols

The header layers of the syntax are considered to be those syntactical elements above the block and macroblock layers (see Figure 6 and the syntax specification in the text). The header levels of the basic Version 1 syntax can form three possible strings, (PSTUF)--PSC--TR--PTYPE--PQUANT--CPM--(PSBI)--(TRB-DBQUANT)--PEI--(PSUPP--PEI--...), (GSTUF)--GBSC--GN--(GSBI)--GFID--GQUANT, and (ESTUF)--(EOS)--(PSTUF). In the revised syntax of Version 2, the header levels of the syntax can have other structures (see Figure 6 and the syntax specification in the text). The header level syntax strings are directly sent to PSC_FIFO as in the normal VLC table mode of H.263 at encoder side, and are directly sent out from PSC_FIFO in the decoder after a legal PSC, GBSC, SSC, EOS, or EOSBS is detected.

If a header is not the first in a video session, the arithmetic encoder needs to be reset before sending the header by calling the following procedure. This procedure shall also be called at the end of a video session if (ESTUF)--EOS (or (ESTUF)--EOSBS for the Sub-Bitstream for which the last header was sent) is not sent.

```
void    encoder_flush( )

{
        opposite_bits++;
        if (low < q1) {
            send out a bit "0" to PSC_FIFO;
            while (opposite_bits > 0) {
                send out a bit "1" to PSC_FIFO;
                opposite_bits--;
            }
        }
        else {
            send out a bit "1" to PSC_FIFO;
            while (opposite_bits > 0) {
                send out a bit "0" to PSC_FIFO;
                opposite_bits--;
            }
        }
        low = 0;
        high = top;
}
```

In the decoder, after each fixed length symbol string, procedure **decoder_reset** is called.

E.7 Macroblock and Block layer symbols

Models for the macroblock and block layer symbols are included in E..8. The indices as given in the VLC tables of clause 5 are used for the indexing of the integers in the models.

The model for COD in P-pictures is named cumf_COD.. The index for COD being "0" is 0, and is 1 for COD being "1". The model for MCBPC in P pictures is named cumf_MCBPC_no4MVQ unless PLUSPTYPE is present in the picture header and either the Advanced Prediction mode (Annex F) or the Deblocking Filter mode (Annex J) is in use, in which case the model is named cumf_MCBPC_4MVQ The indexes for MCBPC are defined in Table 7 for I-pictures and Table 8 for P-pictures. The model for MCBPC in I-pictures is named by cumf_MCBPC_intra.

The model for MODB is cumf_MODB_G if Annex G is used or cumf_MODB_M if Annex M is used. The indexes for MODB are defined in Table 11 or Table M.I, respectively. The model for CBPBn, n=1,2,...,4, is cumf_YCBPB, and the model for CBPBn, n=5,6, is cumf_UVCBPB, with index 0 for CBPBn=0 and index 1 for CBPBn=1.

The model for CBPY is cumf_CBPY in INTER macroblocks and cumf_CBPY_intra in INTRA macroblocks. The model for DQUANT is cumf_DQUANT. The indexing for CBPY and DQUANT is defined in Tables 13 and 12, respectively.

The model for MVD, $MVD_{2\text{-}4}$ and MVDB is cumf_MVD and the model for INTRADC is cumf_INTRADC. The indexing is defined in Tables 14 and 15, respectively.

A non-escaped TCOEF consists of a symbol for TCOEF1/2/3/r followed by a symbol, SIGN, for the sign of the TCOEF. The models for TCOEF1, TCOEF2, TCOEF3 and TCOEFr in INTER blocks are cumf_TCOEF1, cumf_TCOEF2, cumf_TCOEF3, cumf_TCOEFr. The models for INTRA blocks are cumf_TCOEF1_intra, cumf_TCOEF2_intra, cumf_TCOEF3_intra, cumf_TCOEFr_intra. For all TCOEFs the indexing is defined in Table 16. The model for SIGN is cumf_SIGN. The indexing for SIGN is 0 for positive sign and 1 for negative sign.

The models for LAST, RUN, LEVEL after ESCAPE are cumf_LAST (cumf_LAST_intra), cumf_RUN (cumf_RUN_intra), cumf_LEVEL (cumf_LEVEL_intra) for INTER (INTRA) blocks. The indexing for LAST is 0 for LAST=0 and 1 for LAST=1, while the indexing for RUN and LEVEL is defined in Table 17.

The model for INTRA_MODE is cumf_INTRA_AC_DC. The indexing is defined in Table I.1.

E.8 SAC Models

int cumf_COD[3]={16383, 6849, 0};

int cumf_MCBPC_no4MVQ[22]={16383, 4105, 3088, 2367, 1988, 1621, 1612, 1609, 1608, 496, 353, 195, 77, 22, 17, 12, 5, 4, 3, 2, 1, 0};

int cumf_MCBPC_4MVQ[26]={16383, 6880, 6092, 5178, 4916, 3965, 3880, 3795, 3768, 1491, 1190, 889, 655, 442, 416, 390, 360, 337, 334, 331, 327, 326, 88, 57, 26, 0};

int cumf_MCBPC_intra[10]={16383, 7410, 6549, 5188, 442, 182, 181, 141, 1, 0};

int cumf_MODB_G[4]={16383, 6062, 2130, 0};

int cumf_MODB_M[7] = {16383, 6717, 4568, 2784, 1370, 655, 0};

int cumf_YCBPB[3]={16383, 6062, 0};

int cumf_UVCBPB[3]={16383, 491, 0};

int cumf_CBPY[17]={16383, 14481, 13869, 13196, 12568, 11931, 11185, 10814, 9796, 9150, 8781, 7933, 6860, 6116, 4873, 3538, 0};

int cumf_CBPY_intra[17]={16383, 13619, 13211, 12933, 12562, 12395, 11913, 11783, 11004, 10782, 10689, 9928, 9353, 8945, 8407, 7795, 0};

int cumf_DQUANT[5]={16383, 12287, 8192, 4095, 0};

int cumf_MVD[65]={16383, 16380, 16369, 16365, 16361, 16357, 16350, 16343, 16339, 16333, 16326, 16318, 16311, 16306, 16298, 16291, 16283, 16272, 16261, 16249, 16235, 16222, 16207, 16175, 16141, 16094, 16044, 15936, 15764, 15463, 14956, 13924, 11491, 4621, 2264, 1315, 854, 583, 420, 326, 273, 229, 196, 166, 148, 137, 123, 114, 101, 91, 82, 76, 66, 59, 53, 46, 36, 30, 26, 24, 18, 14, 10, 5, 0};

int cumf_INTRADC[255]={16383, 16380, 16379, 16378, 16377, 16376, 16370, 16361, 16360, 16359, 16358, 16357, 16356, 16355, 16343, 16238, 16237, 16236, 16230, 16221, 16220, 16205, 16190, 16169, 16151, 16130, 16109, 16094, 16070, 16037, 16007, 15962, 15938, 15899, 15854, 15815, 15788, 15743, 15689, 15656, 15617, 15560, 15473, 15404, 15296, 15178, 15106, 14992, 14868, 14738, 14593, 14438, 14283, 14169, 14064, 14004, 13914, 13824, 13752, 13671, 13590, 13515, 13458, 13380, 13305, 13230, 13143, 13025, 12935, 12878, 12794, 12743, 12656, 12596, 12521, 12443, 12359, 12278, 12200, 12131, 12047, 12002, 11948, 11891, 11828, 11744, 11663, 11588, 11495, 11402, 11288, 11204, 11126, 11039, 10961, 10883, 10787, 10679, 10583, 10481, 10360, 10227, 10113, 9961, 9828, 9717, 9584, 9485, 9324, 9112, 9019, 8908, 8766, 8584, 8426, 8211, 7920, 7663, 7406, 7152, 6904, 6677, 6453, 6265, 6101, 5904, 5716, 5489, 5307, 5056, 4850, 4569, 4284, 3966, 3712, 3518, 3342, 3206, 3048, 2909, 2773, 2668, 2596, 2512, 2370, 2295, 2232, 2166, 2103, 2022, 1956, 1887, 1830, 1803, 1770, 1728, 1674, 1635, 1599, 1557, 1500, 1482, 1434, 1389, 1356, 1317, 1284, 1245, 1200, 1179, 1140, 1110, 1092, 1062, 1044, 1035, 1014, 1008, 993, 981, 954, 936, 912, 894, 876, 864, 849, 828, 816, 801, 792, 777, 756, 732, 690, 660, 642, 615, 597, 576, 555, 522, 489, 459, 435, 411, 405, 396, 387, 375, 360, 354, 345, 344, 329, 314, 293, 278, 251, 236, 230, 224, 215, 214, 208, 199, 193, 184, 178, 169, 154, 127, 100, 94, 73, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 20, 19, 18, 17, 16, 15, 9, 0};

int cumf_TCOEF1[104]={16383, 13455, 12458, 12079, 11885, 11800, 11738, 11700, 11681, 11661, 11651, 11645, 11641, 10572, 10403, 10361, 10346, 10339, 10335, 9554, 9445, 9427, 9419, 9006, 8968, 8964, 8643, 8627, 8624, 8369, 8354, 8352, 8200, 8192, 8191, 8039, 8036, 7920, 7917, 7800, 7793, 7730, 7727, 7674, 7613, 7564, 7513, 7484, 7466, 7439, 7411, 7389, 7373, 7369, 7359, 7348, 7321, 7302, 7294, 5013, 4819, 4789, 4096, 4073, 3373, 3064, 2674, 2357, 2177, 1975, 1798, 1618, 1517, 1421, 1303, 1194, 1087, 1027, 960, 890, 819, 758, 707, 680, 656, 613, 566, 534, 505, 475, 465, 449, 430, 395, 358, 335, 324, 303, 295, 286, 272, 233, 215, 0};

int cumf_TCOEF2[104]={16383, 13582, 12709, 12402, 12262, 12188, 12150, 12131, 12125, 12117, 12113, 12108, 12104, 10567, 10180, 10070, 10019, 9998, 9987, 9158, 9037, 9010, 9005, 8404, 8323, 8312, 7813, 7743, 7726, 7394, 7366, 7364, 7076, 7062, 7060, 6810, 6797, 6614, 6602, 6459, 6454, 6304, 6303, 6200, 6121, 6059, 6012, 5973, 5928, 5893, 5871, 5847, 5823, 5809, 5796, 5781, 5771, 5763, 5752, 4754, 4654, 4631, 3934, 3873, 3477, 3095, 2758, 2502, 2257, 2054, 1869, 1715, 1599, 1431, 1305, 1174, 1059, 983, 901, 839, 777, 733, 683, 658, 606, 565, 526, 488, 456, 434, 408, 380, 361, 327, 310, 296, 267, 259, 249, 239, 230, 221, 214, 0};

int cumf_TCOEF3[104]={16383, 13532, 12677, 12342, 12195, 12112, 12059, 12034, 12020, 12008, 12003, 12002, 12001, 10586, 10297, 10224, 10202, 10195, 10191, 9223, 9046, 8999, 8987, 8275, 8148, 8113, 7552, 7483, 7468, 7066, 7003, 6989, 6671, 6642, 6631, 6359, 6327, 6114, 6103, 5929, 5918, 5792, 5785, 5672, 5580, 5507, 5461, 5414, 5382, 5354, 5330, 5312, 5288, 5273, 5261, 5247, 5235, 5227, 5219, 4357, 4277, 4272, 3847, 3819, 3455, 3119, 2829, 2550, 2313, 2104, 1881, 1711, 1565, 1366, 1219, 1068, 932, 866, 799, 750, 701, 662, 605, 559, 513, 471, 432, 403, 365, 336, 312, 290, 276, 266, 254, 240, 228, 223, 216, 206, 199, 192, 189, 0};

int cumf_TCOEFr[104]={16383, 13216, 12233, 11931, 11822, 11776, 11758, 11748, 11743, 11742, 11741, 11740, 11739, 10203, 9822, 9725, 9691, 9677, 9674, 8759, 8609, 8576, 8566, 7901, 7787, 7770, 7257, 7185, 7168, 6716, 6653, 6639, 6276, 6229, 6220, 5888, 5845, 5600, 5567, 5348, 5327, 5160, 5142, 5004, 4900, 4798, 4743, 4708, 4685, 4658, 4641, 4622, 4610, 4598, 4589, 4582, 4578, 4570, 4566, 3824, 3757, 3748, 3360, 3338,

3068, 2835, 2592, 2359, 2179, 1984, 1804, 1614, 1445, 1234, 1068, 870, 739, 668, 616, 566, 532, 489, 453, 426, 385, 357, 335, 316, 297, 283, 274, 266, 259, 251, 241, 233, 226, 222, 217, 214, 211, 209, 208, 0};

int cumf_TCOEF1_intra[104]={16383, 13383, 11498, 10201, 9207, 8528, 8099, 7768, 7546, 7368, 7167, 6994, 6869, 6005, 5474, 5220, 5084, 4964, 4862, 4672, 4591, 4570, 4543, 4397, 4337, 4326, 4272, 4240, 4239, 4212, 4196, 4185, 4158, 4157, 4156, 4140, 4139, 4138, 4137, 4136, 4125, 4124, 4123, 4112, 4111, 4110, 4109, 4108, 4107, 4106, 4105, 4104, 4103, 4102, 4101, 4100, 4099, 4098, 4097, 3043, 2897, 2843, 1974, 1790, 1677, 1552, 1416, 1379, 1331, 1288, 1251, 1250, 1249, 1248, 1247, 1236, 1225, 1224, 1223, 1212, 1201, 1200, 1199, 1198, 1197, 1196, 1195, 1194, 1193, 1192, 1191, 1190, 1189, 1188, 1187, 1186, 1185, 1184, 1183, 1182, 1181, 1180, 1179, 0};

int cumf_TCOEF2_intra[104]={16383, 13242, 11417, 10134, 9254, 8507, 8012, 7556, 7273, 7062, 6924, 6839, 6741, 6108, 5851, 5785, 5719, 5687, 5655, 5028, 4917, 4864, 4845, 4416, 4159, 4074, 3903, 3871, 3870, 3765, 3752, 3751, 3659, 3606, 3580, 3541, 3540, 3514, 3495, 3494, 3493, 3474, 3473, 3441, 3440, 3439, 3438, 3425, 3424, 3423, 3422, 3421, 3420, 3401, 3400, 3399, 3398, 3397, 3396, 2530, 2419, 2360, 2241, 2228, 2017, 1687, 1576, 1478, 1320, 1281, 1242, 1229, 1197, 1178, 1152, 1133, 1114, 1101, 1088, 1087, 1086, 1085, 1072, 1071, 1070, 1069, 1068, 1067, 1066, 1065, 1064, 1063, 1062, 1061, 1060, 1059, 1058, 1057, 1056, 1055, 1054, 1053, 1052, 0};

int cumf_TCOEF3_intra[104]={16383, 12741, 10950, 10071, 9493, 9008, 8685, 8516, 8385, 8239, 8209, 8179, 8141, 6628, 5980, 5634, 5503, 5396, 5327, 4857, 4642, 4550, 4481, 4235, 4166, 4151, 3967, 3922, 3907, 3676, 3500, 3324, 3247, 3246, 3245, 3183, 3168, 3084, 3069, 3031, 3030, 3029, 3014, 3013, 2990, 2975, 2974, 2973, 2958, 2943, 2928, 2927, 2926, 2925, 2924, 2923, 2922, 2921, 2920, 2397, 2298, 2283, 1891, 1799, 1591, 1445, 1338, 1145, 1068, 1006, 791, 768, 661, 631, 630, 615, 592, 577, 576, 561, 546, 523, 508, 493, 492, 491, 476, 475, 474, 473, 472, 471, 470, 469, 468, 453, 452, 451, 450, 449, 448, 447, 446, 0};

int cumf_TCOEFr_intra[104]={16383, 12514, 10776, 9969, 9579, 9306, 9168, 9082, 9032, 9000, 8981, 8962, 8952, 7630, 7212, 7053, 6992, 6961, 6940, 6195, 5988, 5948, 5923, 5370, 5244, 5210, 4854, 4762, 4740, 4384, 4300, 4288, 4020, 3968, 3964, 3752, 3668, 3511, 3483, 3354, 3322, 3205, 3183, 3108, 3046, 2999, 2981, 2974, 2968, 2961, 2955, 2949, 2943, 2942, 2939, 2935, 2934, 2933, 2929, 2270, 2178, 2162, 1959, 1946, 1780, 1651, 1524, 1400, 1289, 1133, 1037, 942, 849, 763, 711, 591, 521, 503, 496, 474, 461, 449, 442, 436, 426, 417, 407, 394, 387, 377, 373, 370, 367, 366, 365, 364, 363, 362, 358, 355, 352, 351, 350, 0};

int cumf_SIGN[3]={16383, 8416, 0};

int cumf_LAST[3]={16383, 9469, 0};

int cumf_LAST_intra[3]={16383, 2820, 0};

int cumf_RUN[65]={16383, 15310, 14702, 13022, 11883, 11234, 10612, 10192, 9516, 9016, 8623, 8366, 7595, 7068, 6730, 6487, 6379, 6285, 6177, 6150, 6083, 5989, 5949, 5922, 5895, 5828, 5774, 5773, 5394, 5164, 5016, 4569, 4366, 4136, 4015, 3867, 3773, 3692, 3611, 3476, 3341, 3301, 2787, 2503, 2219, 1989, 1515, 1095, 934, 799, 691, 583, 435, 300, 246, 206, 125, 124, 97, 57, 30, 3, 2, 1, 0};

int cumf_RUN_intra[65]={16383, 10884, 8242, 7124, 5173, 4745, 4246, 3984, 3034, 2749, 2607, 2298, 966, 681, 396, 349, 302, 255, 254, 253, 206, 159, 158, 157, 156, 155, 154, 153, 106, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0};

int cumf_LEVEL[255]={16383, 16382, 16381, 16380, 16379, 16378, 16377, 16376, 16375, 16374, 16373, 16372, 16371, 16370, 16369, 16368, 16367, 16366, 16365, 16364, 16363, 16362, 16361, 16360, 16359, 16358, 16357, 16356, 16355, 16354, 16353, 16352, 16351, 16350, 16349, 16348, 16347, 16346, 16345, 16344, 16343, 16342, 16341, 16340, 16339, 16338, 16337, 16336, 16335, 16334, 16333, 16332, 16331, 16330, 16329, 16328, 16327, 16326, 16325, 16324, 16323, 16322, 16321, 16320, 16319, 16318, 16317, 16316, 16315, 16314, 16313, 16312, 16311, 16310, 16309, 16308, 16307, 16306, 16305, 16304, 16303, 16302, 16301, 16300, 16299, 16298, 16297, 16296, 16295, 16294, 16293, 16292, 16291, 16290, 16289, 16288, 16287, 16286, 16285, 16284, 16283, 16282, 16281, 16280, 16279, 16278, 16277, 16250, 16223, 16222, 16195, 16154, 16153, 16071, 15989, 15880, 15879, 15878, 15824, 15756, 15674, 15606, 15538, 15184, 14572, 13960, 10718, 7994, 5379, 2123, 1537, 992, 693, 611, 516, 448, 421, 380, 353, 352, 284, 257, 230, 203, 162, 161, 160, 133, 132, 105, 104, 103, 102, 101, 100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86, 85, 84, 83, 82, 81, 80, 79, 78, 77, 76, 75, 74, 73, 72, 71, 70, 69, 68, 67, 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0};

int cumf_LEVEL_intra[255]={16383, 16379, 16378, 16377, 16376, 16375, 16374, 16373, 16372, 16371, 16370, 16369, 16368, 16367, 16366, 16365, 16364, 16363, 16362, 16361, 16360, 16359, 16358, 16357, 16356, 16355, 16354, 16353, 16352, 16351, 16350, 16349, 16348, 16347, 16346, 16345, 16344, 16343, 16342, 16341, 16340,

16339, 16338, 16337, 16336, 16335, 16334, 16333, 16332, 16331, 16330, 16329, 16328, 16327, 16326, 16325, 16324, 16323, 16322, 16321, 16320, 16319, 16318, 16317, 16316, 16315, 16314, 16313, 16312, 16311, 16268, 16267, 16224, 16223, 16180, 16179, 16136, 16135, 16134, 16133, 16132, 16131, 16130, 16129, 16128, 16127, 16126, 16061, 16018, 16017, 16016, 16015, 16014, 15971, 15970, 15969, 15968, 15925, 15837, 15794, 15751, 15750, 15749, 15661, 15618, 15508, 15376, 15288, 15045, 14913, 14781, 14384, 13965, 13502, 13083, 12509, 12289, 12135, 11892, 11738, 11429, 11010, 10812, 10371, 9664, 9113, 8117, 8116, 8028, 6855, 5883, 4710, 4401, 4203, 3740, 3453, 3343, 3189, 2946, 2881, 2661, 2352, 2132, 1867, 1558, 1382, 1250, 1162, 1097, 1032, 967, 835, 681, 549, 439, 351, 350, 307, 306, 305, 304, 303, 302, 301, 300, 299, 298, 255, 212, 211, 210, 167, 166, 165, 164, 163, 162, 161, 160, 159, 158, 115, 114, 113, 112, 111, 68, 67, 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0};

int cumf_INTRA_AC_DC[4]={16383, 9229, 5461, 0};

Annex F
Advanced Prediction mode

(This annex forms an integral part of this Recommendation)

F.1 Introduction

This annex describes the optional Advanced Prediction mode of H.263, including overlapped block motion compensation and the possibility of four motion vectors per macroblock. The capability of this mode is signalled by external means (for example Recommendation H.245). The use of this mode is indicated in PTYPE. In the Advanced Prediction mode, motion vectors are allowed to cross picture boundaries as is the case in the Unrestricted Motion Vector mode (for the description of this technique refer to D.1). The extended motion vector range feature of the Unrestricted Motion Vector mode is not automatically included in the Advanced Prediction mode, and only is active if the Unrestricted Motion Vector mode is selected. If the Advanced Prediction mode is used in combination with the PB-frames mode, overlapped motion compensation is only used for prediction of the P-pictures, not for the B-pictures.

F.2 Four motion vectors per macroblock

In this Recommendation, one motion vector per macroblock is used except when in Advanced Prediction mode or Deblocking Filter mode. In this mode, the one/four vectors decision is indicated by the MCBPC codeword for each macroblock. If only one motion vector is transmitted for a certain macroblock, this is defined as four vectors with the same value. If MCBPC indicates that four motion vectors are transmitted for the current macroblock, the information for the first motion vector is transmitted as the codeword MVD and the information for the three additional motion vectors is transmitted as the codewords MVD_{2-4} (see also 5.3.7 and 5.3.8).

The vectors are obtained by adding predictors to the vector differences indicated by MVD and MVD_{2-4} in a similar way as when only one motion vector per macroblock is present, according to the decision rules given in 6.1.1. Again the predictors are calculated separately for the horizontal and vertical components. However, the candidate predictors MV1, MV2 and MV3 are redefined as indicated in Figure F.1. If only one vector per macroblock is present, MV1, MV2 and MV3 are defined as for the 8 * 8 block numbered 1 in Figure 5 (this definition is given in the upper left of the four sub-figures of Figure F.1).

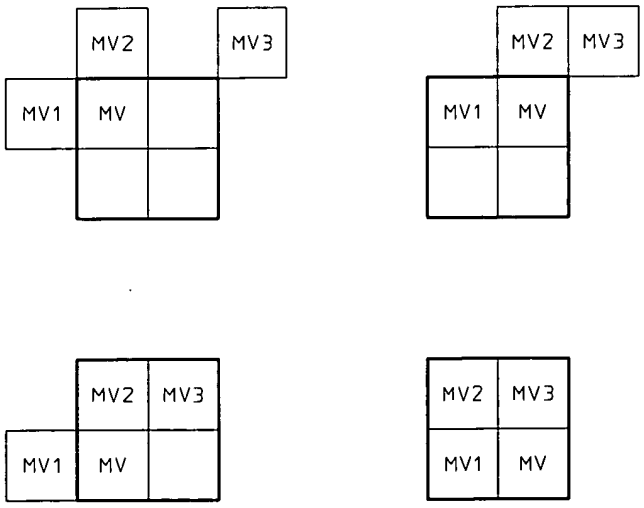


FIGURE F.1/H.263

Redefinition of the candidate predictors MV1, MV2 and MV3

for each of the luminance blocks in a macroblock

If four vectors are used, each of the motion vectors is used for all pixels in one of the four luminance blocks in the macroblock. The numbering of the motion vectors is equivalent to the numbering of the four luminance blocks as given in Figure 5. Motion vector MVD_{CHR} for both chrominance blocks is derived by calculating the sum of the four luminance vectors and dividing this sum by 8; the component values of the resulting sixteenth pixel resolution vectors are modified towards the nearest half pixel position as indicated in Table F.1.

TABLE F.1/H.263

Modification of sixteenth pixel resolution chrominance vector components

sixteenth pixel position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	/16
resulting position	0	0	0	1	1	1	1	1	1	1	1	1	1	1	2	2	/2

Half pixel values are found using bilinear interpolation as described in 6.1.2. In Advanced Prediction mode, the prediction for luminance is obtained by overlapped motion compensation as described in Section F.3. The prediction for chrominance is obtained by applying the motion vector MVD_{CHR} to all pixels in the two chrominance blocks (as it is done in the default prediction mode).

F.3 Overlapped motion compensation for luminance

Each pixel in an 8 * 8 luminance prediction block is a weighted sum of three prediction values, divided by 8 (with rounding). In order to obtain the three prediction values, three motion vectors are used: the motion vector of the current luminance block, and two out of four "remote" vectors:

- the motion vector of the block at the left or right side of the current luminance block;
- the motion vector of the block above or below the current luminance block.

For each pixel, the remote motion vectors of the blocks at the two nearest block borders are used. This means that for the upper half of the block the motion vector corresponding to the block above the current block is used, while for the lower half of the block the motion vector corresponding to the block below the current block is used (see Figure F.3). Similarly, for the left half of the block the motion vector corresponding to the block at the left side of the current block is used, while for the right half of the block the motion vector corresponding to the block at the right side of the current block is used (see Figure F.4).

Let (x,y) be a position in a picture measured in integer pixel units.

Let (m,n) be an integer block index in a picture, as given by

$$m = x / 8 \quad \text{and} \quad n = y / 8,$$

where "/" denotes division with truncation.

Let (i,j) be an integer pixel location in an 8×8 block, given by

$$i = x - m \cdot 8 \quad \text{and} \quad j = y - n \cdot 8,$$

resulting in

$$(x,y) = (m \cdot 8 + i, n \cdot 8 + j).$$

Let (MV^k_x, MV^k_y) be a motion vector, which may contain full-pixel or half pixel offsets, with $k = 0$, 1, or 2. For example, (MV^k_x, MV^k_y) can be equal to (-7.0, 13.5). Here, (MV^0_x, MV^0_y) denotes the motion vector for the current block (m,n), (MV^1_x, MV^1_y) denotes the motion vector of the block either above or below, and (MV^2_x, MV^2_y) denotes the motion vector either to the left or right of the current block (m,n) as defined above.

Then the creation of each pixel, $P(x,y)$, in an 8×8 luminance prediction block with block index (m,n) is governed by the following equation:

$$P(x,y) = (q(x,y) \cdot H_0(i,j) + r(x,y) \cdot H_1(i,j) + s(x,y) \cdot H_2(i,j) + 4)/8,$$

where $q(x,y)$, $r(x,y)$ and $s(x,y)$ are the prediction values taken from the referenced picture as defined by

$q(x,y) = p(x + MV^0_x, y + MV^0_y)$,

$r(x,y) = p(x + MV^1_x, y + MV^1_y)$,

$s(x,y) = p(x + MV^2_x, y + MV^2_y)$,

where $p(x+MV^k_x, y+MV^k_y)$ is the prediction value at position $(x+MV^k_x, y+MV^k_y)$ in the referenced picture. Note that $(x+MV^k_x, y+MV^k_y)$ may be outside the picture and can be at a full or half pixel position. In cases using half pixel motion vectors, $p(x+MV^k_x, y+MV^k_y)$ refers to the value obtained after the interpolation process described in section 6.1.2 is applied.

The matrices $H_0(i,j)$, $H_1(i,j)$ and $H_2(i,j)$ are defined in Figures F.2, F.3 and F.4, where (i, j) denotes the column and row, respectively, of the matrix.

When neither the Slice Structured mode (see Annex K) nor the Independent Segment Decoding mode (see Annex R) are in use, remote motion vectors from other video picture segments are used in the same way as remote motion vectors inside the current GOB. If either the Slice Structured mode or the Independent Segment Decoding mode are in use, the remote motion vectors corresponding to blocks from other video picture segments are set to the motion vector of the current block, regardless of the other conditions described in the next paragraph. (See Annex R for the definition of a video picture segment.)

If one of the surrounding macroblocks was not coded, the corresponding remote motion vector is set to zero. If one of the surrounding blocks was INTRA coded, the corresponding remote motion vector is replaced by the motion vector for the current block except when in PB-frames mode. In this case (INTRA block in PB-frame mode), the INTRA block's motion vector is used (also see Annex G). If the current block is at the border of the picture and therefore a surrounding block is not present, the corresponding remote motion vector is replaced by the current motion vector. In all cases, if the current block is at the bottom of the macroblock (for block number 3 or 4, see Figure 5), the remote motion vector corresponding with an 8 * 8 luminance block in the macroblock below the current macroblock is replaced by the motion vector for the current block.

The weighting values for the prediction are given in Figures F.2, F.3 and F.4.

4	5	5	5	5	5	5	4
5	5	5	5	5	5	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	5	5	5	5	5	5
4	5	5	5	5	5	5	4

FIGURE F.2/H.263

Weighting values, H_0, for prediction with motion vector of current luminance block

2	2	2	2	2	2	2	2
1	1	2	2	2	2	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	2	2	2	2	1	1
2	2	2	2	2	2	2	2

FIGURE F.3/H.263

Weighting values, H_1, for prediction with motion vectors of the luminance blocks on top or bottom of current luminance block

2	1	1	1	1	1	1	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	1	1	1	1	1	1	2

FIGURE F.4/H.263

Weighting values, H_2, for prediction with motion vectors of the luminance blocks to the left or right of current luminance block

Annex G

PB-frames mode

(This annex forms an integral part of this Recommendation)

G.1 Introduction

This annex describes the optional PB-frames mode of H.263. The capability of this mode is signalled by external means (for example Recommendation H.245). The use of this mode is indicated in PTYPE.

A PB-frame consists of two pictures being coded as one unit. The name PB comes from the name of picture types in Recommendation H.262 where there are P-pictures and B-pictures. Thus a PB-frame consists of one P-picture which is predicted from the previous decoded P-picture and one B-picture which is predicted both from the previous decoded P-picture and the P-picture currently being decoded. The name B-picture was chosen because parts of B-pictures may be bidirectionally predicted from the past and future pictures. The prediction process is illustrated in Figure G.1.

An improved version of the PB frames mode, termed the "Improved PB frames mode," is described in Annex M. The PB frames mode as described in this annex is retained herein only for purposes of compatibility with systems designed prior to the adoption of the Improved PB frames mode. For this reason, the PB frames mode as described in this annex cannot be used with the additional features of the syntax which require the use of PLUSPTYPE.

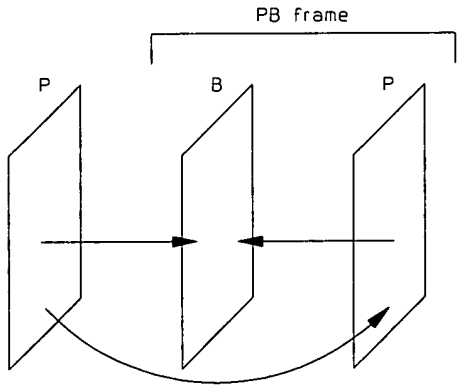


FIGURE G.1/H.263

Prediction in PB-frames mode

G.2 PB-frames and INTRA blocks

When PB-frames are used the coding mode INTRA has the following meaning (see also 5.3.2):

- The P-blocks are INTRA coded.
- The B-blocks are INTER coded with prediction as for an INTER block.

If PB-frames are used, Motion Vector Data (MVD) is included also for INTRA macroblocks in pictures for which PTYPE indicates "INTER". In this case the vector is used for the B-blocks only. The codewords MVD_{2-4} are never used for INTRA (see also Table 10). When in both the Advanced Prediction mode and the PB-frames mode, and

one of the surrounding blocks was coded in INTRA mode, the corresponding remote motion vector is not replaced by the motion vector for the current block. Instead, the remote "INTRA" motion vector is used.

G.3 Block Layer

In a PB-frame, a macroblock comprises twelve blocks. First the data for the six P-blocks is transmitted as in the default H.263 mode, then the data for the six B-blocks (see also 5.4). The structure of the block layer is shown in Figure 11. INTRADC is present for every P-block of the macroblock if MCBPC indicates macroblock type 3 or 4 (see Tables 7 and 8). INTRADC is not present for B-blocks. TCOEF is present for P-blocks if indicated by MCBPC or CBPY; TCOEF is present for B-blocks if indicated by CBPB.

G.4 Calculation of vectors for the B-picture in a PB-frame

The vectors for the B-picture are calculated as follows (see also 6.1.1). Assume we have a vector component MV in half pixel units to be used in the P-picture (MV represents a vector component for an 8 * 8 luminance block; if only one vector per macroblock is transmitted, MV has the same value for each of the four 8 * 8 luminance blocks). For prediction of the B-picture we need both forward and backward vector components MV_F and MV_B. These forward and backward vector components are derived from MV and eventually enhanced by a delta vector given by MVDB.

- TR_D The increment of the temporal reference TR (or the combined extended temporal reference ETR and temporal reference TR in an Improved PB frame when a custom picture clock frequency is in use) from the last picture header (see 5.1.2). If TR_D is negative then $TR_D = TR_D + d$ where d = 256 for CIF picture frequency and 1024 for any custom picture clock frequency
- TR_B See 5.1.21.

Assume that MV_D is the delta vector component given by MVDB and corresponding with vector component MV. If MVDB is not present, MV_D is set to zero. If MVDB is present, the same MV_D given by MVDB is used for each of the four luminance B-blocks within the macroblock.

Now MV_F and MV_B are given in half pixel units by the following formulas:

$$MV_F = (TR_B \times MV) / TR_D + MV_D$$

$$MV_B = ((TR_B - TR_D) \times MV) / TR_D \quad \text{if } MV_D \text{ is equal to 0}$$

$$MV_B = MV_F - MV \quad \text{if } MV_D \text{ is unequal to 0}$$

where "/" means division by truncation. It is assumed that the scaling reflects the actual position in time of P- and B-pictures. Advantage is taken of the fact that the range of values for MV_F is constrained. Each VLC word for MVDB represents a pair of difference values. Only one of the pair will yield a value for MV_F falling within the permitted range (default [–16, 15.5]; in Unrestricted Motion Vector mode [–31.5, 31.5]). The formulas for MV_F and MV_B are also used in the case of INTRA blocks where the vector data is used only for predicting B-blocks.

For chrominance blocks, MV_F is derived by calculating the sum of the four corresponding luminance MV_F vectors and dividing this sum by 8; the resulting sixteenth pixel resolution vector components are modified towards the nearest half pixel position as indicated in Table F.1. MV_B for chrominance is derived by calculating the sum of the four corresponding luminance MV_B vectors and dividing this sum by 8; the resulting sixteenth pixel resolution vector components are modified towards the nearest half pixel position as indicated in Table F.1.

A positive value of the horizontal or vertical component of the motion vector signifies that the prediction is formed from pixels in the referenced picture which are spatially to the right or below the pixels being predicted.

G.5 Prediction of a B-block in a PB-frame

In this subclause a block means an 8 × 8 block. The following procedure applies for luminance as well as chrominance blocks. First, the forward and backward vectors are calculated. It is assumed that the P-macroblock (luminance and chrominance) is first decoded, reconstructed and clipped (see 6.3.2). This macroblock is called P_{REC}. Based on P_{REC} and the prediction for P_{REC}, the prediction for the B-block is calculated.

The prediction of the B-block has two modes that are used for different parts of the block:

- For pixels where the backward vector – MV_B – points inside P_{REC}, use bidirectional prediction. This is obtained as the average of the forward prediction using MV_F relative to the previous decoded picture, and the backward prediction using MV_B relative to P_{REC}. The average is calculated by dividing the sum of the two predictions by two (division by truncation).
- For all other pixels, forward prediction using MV_F relative to the previous decoded picture is used.

Figure G.2 indicates which part of a block is predicted bidirectionally (shaded part of the B-block) and which part with forward prediction only (rest of the B-block).

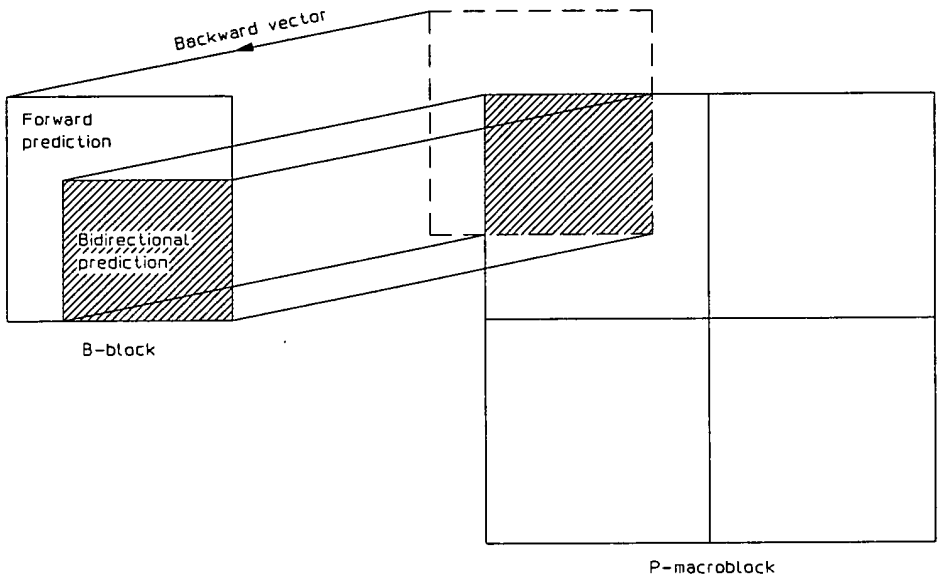


FIGURE G.2/H.263

Forward and bidirectional prediction for a B-block

Bi-directional prediction is used for pixels where the backward vector – MV_B – points inside P_{REC}. These pixels are defined by the following procedures which are specified in C language.

Definitions:

nh	Horizontal position of block within a macroblock (0 or 1).
nv	Vertical position of block within a macroblock (0 or 1).
mh(nh,nv)	Horizontal vector component of block (nh,nv) in half pixel units.
mv(nh,nv)	Vertical vector component of block (nh,nv) in half pixel units.
mhc	Horizontal chrominance vector component.
mvc	Vertical chrominance vector component.

Procedure for luminance

```
for (nh = 0; nh <= 1; nh++) {
   for (nv = 0; nv <= 1; nv++) {
      for (i = nh * 8 + max(0,(–mh(nh,nv)+1)/2 – nh * 8);
            i <= nh * 8 + min(7,15–(mh(nh,nv)+1)/2 – nh * 8); i++) {
         for (j = nv * 8 + max(0,(–mv(nh,nv)+1)/2 – nv * 8);
            j <= nv * 8 + min(7,15–(mv(nh,nv)+1)/2 – nv * 8); j++) {
            predict pixel (i,j) bidirectionally
         }
      }
   }
}
```

Procedure for chrominance

```
for (i = max(0, (–mhc+1)/2); i <= min(7,7–(mhc+1)/2); i++) {
   for (j = max(0, (–mvc+1)/2); j <= min(7,7–(mvc+1)/2); j++) {
      predict pixel (i,j) bidirectionally;
   }
}
```

Pixels not predicted bidirectionally are predicted with forward prediction only.

Annex H

Forward Error Correction for coded video signal

(This annex forms an integral part of this Recommendation)

H.1 Introduction

This annex describes an optional forward error correction method (code and framing) for transmission of H.263 encoded video data. This forward error correction may be used in situations where no forward error correction is provided by external means, for example at the multiplex or system level. It is not used for Recommendation H.324. Both the framing and the forward error correction code are the same as in Recommendation H.261.

H.2 Error correction framing

To allow the video data and error correction parity information to be identified by a decoder an error correction framing pattern is included. This pattern consists of multiframes of eight frames, each frame comprising 1 bit framing, 1 bit fill indicator (Fi), 492 bits of coded data (or fill all 1s) and 18 bits parity (see Figure H.1). For each multiframe the frame alignment pattern formed by the framing bits of the eight individual frames is:

$$(S_1S_2S_3S_4S_5S_6S_7S_8) = (00011011)$$

The fill indicator (Fi) can be set to zero by an encoder. In this case 492 consecutive fill bits (fill all 1s) are used instead of 492 bits of coded data. This may be used for stuffing data (see 3.6).

H.3 Error correcting code

The error correction code is a BCH (511,493) forward error correction code. Use of this by the decoder is optional. The parity is calculated against a code of 493 bits, comprising 1 bit fill indicator (Fi) and 492 bits of coded video data.

The generator polynomial is:

$$g(x) = (x9 + x4 + 1)(x9 + x6 + x4 + x3 + 1)$$

Example: for the input data of "01111 . . . 11" (493 bits) the resulting correction parity bits are "011011010100011011" (18 bits).

H.4 Relock time for error corrector framing

Three consecutive error correction frame alignment patterns (24 bits) should be received before frame lock is deemed to have been achieved. The decoder should be designed such that frame lock will be re-established within 34 000 bits after an error corrector framing phase change.

NOTE – This assumes that the video data does not contain three correctly phased emulations of the error correction framing sequence during the relocking period.

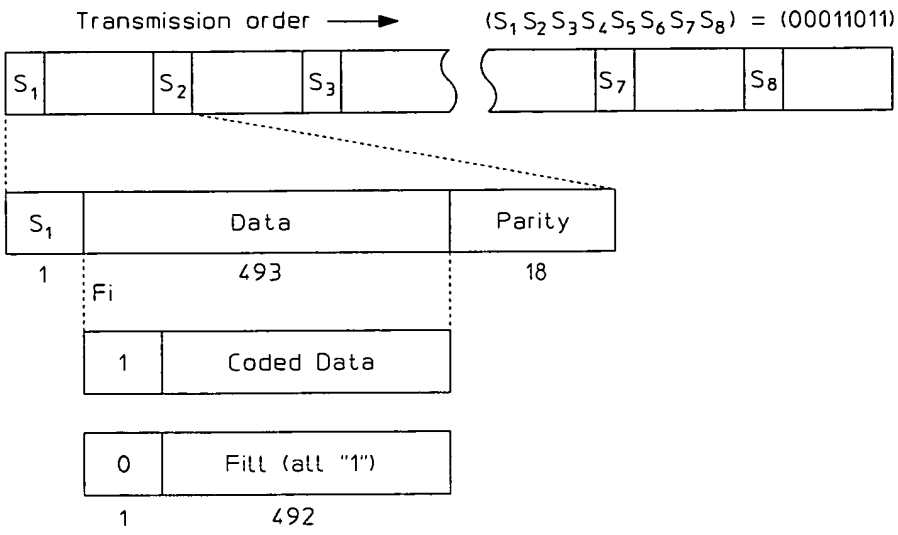


FIGURE H.1/H.263

Error correcting frame

Annex I
Advanced INTRA Coding mode

(This annex forms an integral part of this Recommendation.)

This annex describes the optional Advanced INTRA Coding mode of H.263. The capability of this mode of H.263 is signaled by external means (for example Recommendation H.245). The use of this mode is indicated in the PLUSPTYPE field of the picture header.

I.1 Introduction

This optional mode alters the decoding of macroblocks of type 'INTRA' (macroblocks of other types are not affected). The coding efficiency of INTRA macroblocks is improved by using

1. INTRA-block prediction using neighboring INTRA blocks for the same component (Y, C_B, or C_R),
2. Modified inverse quantization for INTRA coefficients, and
3. A separate VLC for INTRA coefficients.

A particular INTRA-coded block may be predicted from the block above the current block being decoded, from the block to the left of the current block being decoded, or from both. Special cases exist for situations in which the neighboring blocks are not INTRA coded or are not in the same video picture segment. Block prediction always uses data from the same luminance or color difference component (Y, C_B, or C_R) as the block being decoded. In prediction, DC coefficients are always predicted in some manner. The first row of AC coefficients may be predicted from those in the block above, or the first column of AC coefficients may be predicted from those in the block to the left, or only the DC coefficient may be predicted as an average from the block above and the block to the left, as signaled on a macroblock-by-macroblock basis. The remaining AC coefficients are never predicted. Inverse quantization of the INTRADC coefficient is modified to allow a varying quantization step size, unlike in the main text of this Recommendation where a fixed step size of 8 is used for INTRADC coefficients. Inverse quantization of all INTRA coefficients is performed without a "dead-zone" in the quantizer reconstruction spacing.

I.2 Syntax

When using the Advanced INTRA Coding mode, the macroblock layer syntax is altered as specified in Figure I.1. The syntax as shown in Figure I.1 is the same as that defined in section 5.3 except for the insertion of an additional INTRA_MODE field for INTRA macroblocks. INTRA_MODE is present only when MCBPC indicates a macroblock of type INTRA (macroblock type 3 or 4). The prediction mode is coded using the variable length code shown in Table I.1. One prediction mode is transmitted per INTRA macroblock.

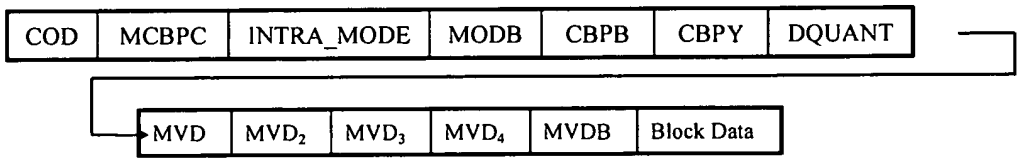


FIGURE I.1/H.263

Structure of macroblock layer

TABLE I.1/H.263

VLC for INTRA_MODE

Index	Prediction Mode	VLC
0	0 (DC Only)	0
1	1 (Vertical DC & AC)	10
2	2 (Horizontal DC & AC)	11

I.3 Decoding Process

Two scans in addition to the zigzag scan are employed. The two added scans are shown in Figure I.2 parts (a) and (b), and the zigzag scan is shown in Figure 14.

For INTRA coded blocks, if Prediction Mode=0, the zigzag scan shown in Figure 14 is selected for all blocks in the macroblock, otherwise, the prediction direction is used to select a scan for the macroblock.

Prediction Mode=1 uses the vertically adjacent block in forming a prediction. This prediction mode is designed for INTRA blocks which are dominated by stronger horizontal frequency content, so the vertically adjacent block is used to predict the horizontal frequency content of the current block, with a prediction of zero for all coefficients representing vertical AC content. Then the scanning pattern is chosen to scan the stronger horizontal frequencies prior to the vertical ones, using the Alternate-Horizontal scan.

Prediction Mode=2 uses the horizontally adjacent block in forming a prediction. This prediction mode is designed for INTRA blocks which are dominated by stronger vertical frequency content, so the horizontally adjacent block is used to predict the vertical frequency content of the current block, with a prediction of zero for all coefficients representing horizontal AC content. Then the scanning pattern is chosen to scan the stronger vertical frequencies prior to the horizontal ones, using the Alternate-Vertical scan.

For non INTRA blocks, the 8x8 blocks of transform coefficients are scanned with "zigzag" scanning as shown in Figure 14.

1	2	3	4	11	12	13	14
5	6	9	10	18	17	16	15
7	8	20	19	27	28	29	30
21	22	25	26	31	32	33	34
23	24	35	36	43	44	45	46
37	38	41	42	47	48	49	50
39	40	51	52	57	58	59	60
53	54	55	56	61	62	63	64

(a) Alternate-Horizontal scan

1	5	7	21	23	37	39	53
2	6	8	22	24	38	40	54
3	9	20	25	35	41	51	55
4	10	19	26	36	42	52	56
11	18	27	31	43	47	57	61
12	17	28	32	44	48	58	62
13	16	29	33	45	49	59	63
14	15	30	34	46	50	60	64

(b) Alternate-Vertical scan (as in Recommendation H.262)

FIGURE I.2/H.263

Alternate DCT Scanning patterns for Advanced INTRA Coding

A separate VLC table is used for all INTRADC and INTRA AC coefficients. This table is specified in Table I.2. Note that the VLC codeword entries used in Table I.2 are the same as those used in the normal TCOEF table (Table 16) used when Advanced INTRA Coding is not in use, but with a different interpretation of LEVEL and RUN (without altering LAST).

Depending on the value of INTRA_MODE either one or eight coefficients are prediction residuals that must be added to a predictor as described below. Figure I.3 shows three 8×8 blocks of final reconstructed DCT levels for the same component (Y, C_B, or C_R), labeled RecA'(u,v), RecB'(u,v) and RecC'(u,v), where u and v are column (horizontal) and row (vertical) indices, respectively. The reconstruction process differs from the processing

described in section 6.2.1. INTRADC residuals are reconstructed differently by use of a variable step size, as opposed to using Table 15, and then a predictor is added to the residual values to obtain the final coefficient reconstruction value. INTRA coefficients other than INTRADC are also reconstructed differently than in section 6.2.1, using a reconstruction spacing without a "dead-zone" and then in some cases adding a predictor to obtain the final coefficient reconstruction value. The block may contain both DC and AC prediction residuals.

The definitions of MCPBC and CBPY are changed when Advanced INTRA Coding is in use. When Advanced INTRA Coding is in use, INTRADC transform coefficients are no longer handled as a separate case, but are instead treated in the same way as the AC coefficients in regard to MCPBC and CBPY. This means that a zero INTRADC will not be coded as a LEVEL, but will simply increase the run for the following AC coefficients.

The inverse quantization process for the B part of an Improved PB frame (see Annex M) is not altered by the use of the Advanced INTRA Coding mode.

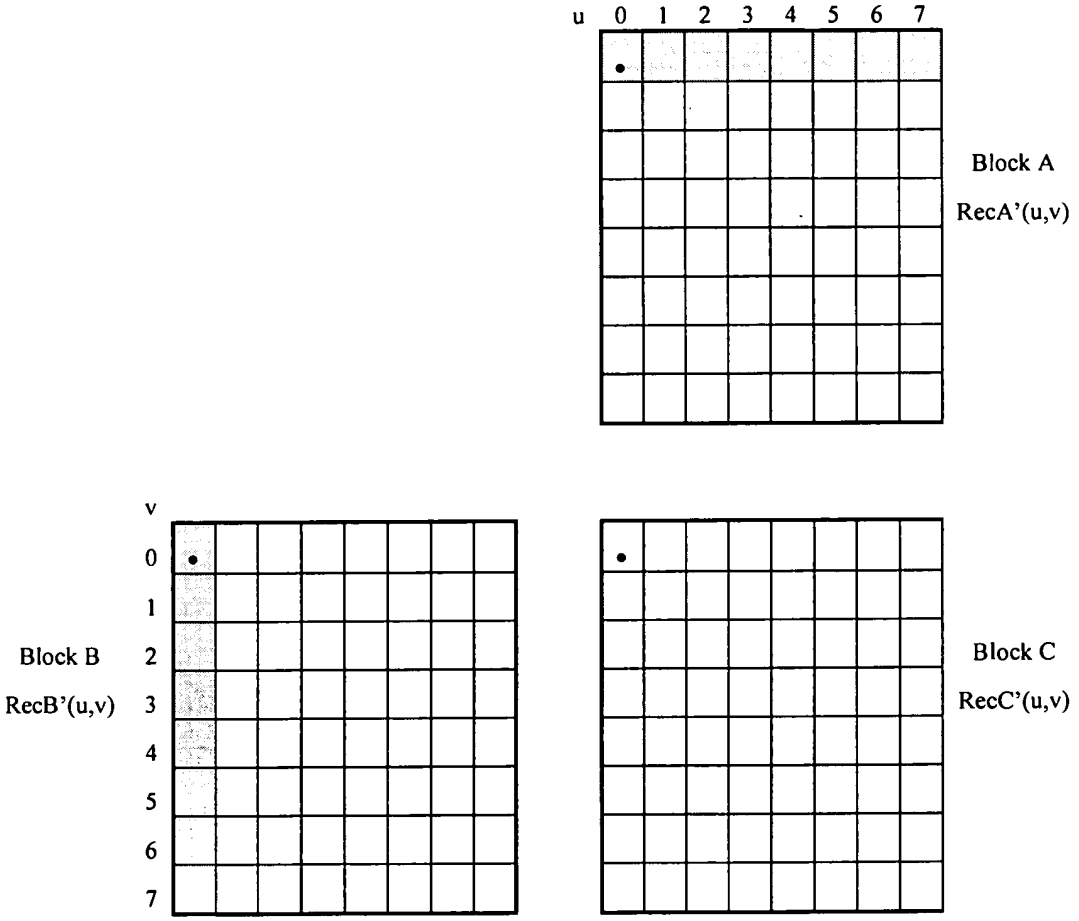


FIGURE I.3/H.263

Three Neighboring blocks in the DCT domain

Define RecC(u,v) to be the reconstructed coefficient *residuals* of the current block. For all INTRA coefficients, the reconstructed residual value is obtained by

$$RecC(u,v) = 2 * QUANT * LEVEL(u,v) \qquad u=0, \ldots, 7, v=0, \ldots, 7.$$

(Note that LEVEL(u,v) denotes a quantity having both a magnitude and a sign in the above equation.)

Define RecC'(u,v) to be the *final* reconstructed coefficient values of the current block (after adjustments for prediction, oddification as described below, and clipping). The final reconstructed coefficient values RecC'(u,v) are

recovered by adding RecC(u,v) to the appropriate prediction as signaled in the INTRA_MODE field, altering the least-significant bit if needed for oddification of the DC coefficient, and clipping.

RecA'(u,v) denotes the *final* reconstructed coefficient values for the block immediately above the current block. RecB'(u,v) denotes the *final* reconstructed coefficient values of the block immediately to the left of the current block.

The ability to use the reconstructed coefficient values for blocks A and B in the prediction of the coefficient values for block C depends on whether blocks A and B are in the same video picture segment as block C. A block is defined to be "in the same video picture segment" as another block only if the following conditions are fulfilled:

1. The relevant block is within the boundary of the picture, and
2. If not in Slice Structured mode (see Annex K), the relevant block is either within the same GOB or no GOB header is present for the current GOB, and
3. If in Slice Structured mode, the relevant block is within the same slice.

The block C to be decoded is predicted only from INTRA blocks within the same video picture segment as block C, as shown below.

If Prediction Mode = 0 is used (DC prediction only) and blocks A and B are both INTRA blocks within the same video picture segment as block C, then the DC coefficient of block C is predicted from the average (with truncation) of the DC coefficients of blocks A and B. If only one of the two blocks A and B is an INTRA block within the same video picture segment as block C, then the DC coefficient of only this one of the two blocks is used as the predictor for Prediction Mode = 0. If neither of the two blocks A and B are INTRA blocks within the same video picture segment as block C, then the prediction uses the value of 1024 as the predictor for the DC coefficient.

If Prediction Mode = 1 or 2 (vertical DC & AC or horizontal DC & AC prediction) and the referenced block (block A or block B) is not an INTRA block within the same video picture segment as block C, then the prediction uses the value of 1024 as the predictor for the DC coefficient and the value of 0 as the predictor for the AC coefficients of block C.

A process of "oddification" is applied to the DC coefficient in order to minimize the impact of IDCT mismatch errors. Certain values of coefficients can cause a round-off error mismatch between different IDCT implementations, especially certain values of the (0,0), (0,4), (4,0), and (4,4) coefficients. For example, a DC coefficient of 8k+4 for some integer k results in an inverse-transformed block having a constant value k+0.5, for which slight errors can cause rounding in different directions on different implementations.

Define the clipAC() function to indicate clipping to the range -2048 to 2047. Define the clipDC() function to indicate clipping to the range 0 to 2047. Define the oddifyclipDC(x) function as

```
If (x is even) {
    result = clipDC(x+1)
} else {
    result = clipDC(x)
}
```

The reconstruction for each INTRA prediction mode is then specified as follows, in which the operator "/" is defined as division by truncation:

mode 0: DC prediction only.

```
RecC'(u,v) = clipAC( RecC(u,v) )                    (u,v) ≠ (0,0), u = 0, ..., 7, v = 0, ..., 7.
If (block A and block B are both INTRA coded and are both in the same video picture segment as block C) {
  tempDC = RecC(0,0) + ( RecA'(0,0) + RecB'(0,0) ) / 2
} else {
  If (block A is INTRA coded and is in the same video picture segment as block C) {
    tempDC = RecC(0,0) + RecA'(0,0)
  } else {
    If (block B is INTRA coded and is in the same video picture segment as block C) {
      tempDC = RecC(0,0) + RecB'(0,0)
    } else {
      tempDC = RecC(0,0) + 1024
    }
  }
}
```

RecC'(0,0) = oddifyclipDC(tempDC)

mode 1: DC and AC prediction from the block above.

If (block A is INTRA coded and is in the same video picture segment as block C) {
  tempDC = RecC(0,0) + RecA'(0,0)
  RecC'(u,0) = clipAC(RecC(u,0) + RecA'(u,0)) u = 1, ..., 7,
  RecC'(u,v) = clipAC(RecC(u,v)) u = 0, ..., 7, v = 1, ..., 7.
} else {
  tempDC = RecC(0,0) + 1024
  RecC'(u,v) = clipAC(RecC(u,v)) (u,v) ≠ (0,0), u = 0, ...,7, v = 0, ..., 7
}
RecC'(0,0) = oddifyclipDC(tempDC)

mode 2: DC and AC prediction from the block to the left.

If (block B is INTRA coded and is in the same video picture segment as block C) {
  tempDC = RecC(0,0) + RecB'(0,0)
  RecC'(0,v) = clipAC(RecC(0,v) + RecB'(0,v)) v = 1, ..., 7,
  RecC'(u,v) = clipAC(RecC(u,v)) u = 1, ..., 7, v = 0, ..., 7.
} else {
  tempDC = RecC(0,0) + 1024
  RecC'(u,v) = clipAC(RecC(u,v)) (u,v) ≠ (0,0), u = 0, ..., 7, v = 0, ..., 7
}
RecC'(0,0) = oddifyclipDC(tempDC)

TABLE I.2/H.263

VLC for INTRA TCOEF

INDEX	LAST	RUN	|LEVEL|	BITS	VLC CODE
0	0	0	1	3	10s
1	0	1	1	5	1111 s
2	0	3	1	7	0101 01s
3	0	5	1	8	0010 111s
4	0	7	1	9	0001 1111 s
5	0	8	1	10	0001 0010 1s
6	0	9	1	10	0001 0010 0s
7	0	10	1	11	0000 1000 01s
8	0	11	1	11	0000 1000 00s
9	0	4	3	12	0000 0000 111s
10	0	9	2	12	0000 0000 110s
11	0	13	1	12	0000 0100 000s
12	0	0	2	4	110s
13	0	1	2	7	0101 00s
14	0	1	4	9	0001 1110 s
15	0	1	5	11	0000 0011 11s
16	0	1	6	12	0000 0100 001s
17	0	1	7	13	0000 0101 0000s
18	0	0	3	5	1110 s
19	0	3	2	9	0001 1101 s
20	0	2	3	11	0000 0011 10s
21	0	3	4	13	0000 0101 0001s
22	0	0	5	6	0110 1s
23	0	4	2	10	0001 0001 1s
24	0	3	3	11	0000 0011 01s
25	0	0	4	6	0110 0s
26	0	5	2	10	0001 0001 0s
27	0	5	3	13	0000 0101 0010s
28	0	2	1	6	0101 1s
29	0	6	2	11	0000 0011 00s
30	0	0	25	13	0000 0101 0011s
31	0	4	1	7	0100 11s
32	0	7	2	11	0000 0010 11s

INDEX	LAST	RUN	|LEVEL|	BITS	VLC CODE
58	1	0	1	5	0111 s
59	1	14	1	10	0000 1100 1s
60	1	20	1	12	0000 0000 101s
61	1	1	1	7	0011 11s
62	1	19	1	12	0000 0000 100s
63	1	2	1	7	0011 10s
64	1	3	1	7	0011 01s
65	1	0	2	7	0011 00s
66	1	5	1	8	0010 011s
67	1	6	1	8	0010 010s
68	1	4	1	8	0010 001s
69	1	0	3	8	0010 000s
70	1	9	1	9	0001 1010 s
71	1	10	1	9	0001 1001 s
72	1	11	1	9	0001 1000 s
73	1	12	1	9	0001 0111 s
74	1	13	1	9	0001 0110 s
75	1	8	1	9	0001 0101 s
76	1	7	1	9	0001 0100 s
77	1	0	4	9	0001 0011 s
78	1	17	1	10	0000 1100 0s
79	1	18	1	10	0000 1011 1s
80	1	16	1	10	0000 1011 0s
81	1	15	1	10	0000 1010 1s
82	1	2	2	10	0000 1010 0s
83	1	1	2	10	0000 1001 1s
84	1	0	6	10	0000 1001 0s
85	1	0	5	10	0000 1000 1s
86	1	4	2	11	0000 0001 11s
87	1	3	2	11	0000 0001 10s
88	1	1	3	11	0000 0001 01s
89	1	0	7	11	0000 0001 00s
90	1	2	3	12	0000 0100 100s

(continued below)

TABLE I.2/H.263 (Continued)

VLC for INTRA TCOEF (Continued)

INDEX	LAST	RUN	LEVEL	BITS	VLC CODE
33	0	0	24	13	0000 0101 0100s
34	0	0	8	7	0100 10s
35	0	8	2	11	0000 0010 10s
36	0	0	7	7	0100 01s
37	0	2	4	11	0000 0010 01s
38	0	0	6	7	0100 00s
39	0	12	1	11	0000 0010 00s
40	0	0	9	8	0010 110s
41	0	0	23	13	0000 0101 0101s
42	0	2	2	8	0010 101s
43	0	1	3	8	0010 100s
44	0	6	1	9	0001 1100 s
45	0	0	10	9	0001 1011 s
46	0	0	12	10	0001 0000 1s
47	0	0	11	10	0001 0000 0s
48	0	0	18	10	0000 1111 1s
49	0	0	17	10	0000 1111 0s
50	0	0	16	10	0000 1110 1s
51	0	0	15	10	0000 1110 0s
52	0	0	14	10	0000 1101 1s
53	0	0	13	10	0000 1101 0s
54	0	0	20	12	0000 0100 010s
55	0	0	19	12	0000 0100 011s
56	0	0	22	13	0000 0101 0110s
57	0	0	21	13	0000 0101 0111s

INDEX	LAST	RUN	LEVEL	BITS	VLC CODE
91	1	1	4	12	0000 0100 101s
92	1	0	9	12	0000 0100 110s
93	1	0	8	12	0000 0100 111s
94	1	21	1	13	0000 0101 1000s
95	1	22	1	13	0000 0101 1001s
96	1	23	1	13	0000 0101 1010s
97	1	7	2	13	0000 0101 1011s
98	1	6	2	13	0000 0101 1100s
99	1	5	2	13	0000 0101 1101s
100	1	3	3	13	0000 0101 1110s
101	1	0	10	13	0000 0101 1111s
102	ESCAPE			7	0000 011

Annex J
Deblocking Filter mode

(This annex forms an integral part of this Recommendation.)

J.1 Introduction

This annex describes the use of an optional block edge filter within the coding loop. The main purpose of the block edge filter is to reduce blocking artifacts. The filtering is performed on 8x8 block edges. Motion vectors may have either 8x8 or 16x16 resolution (see section J.2). The processing described in this annex applies only for the P, I, EP, or EI pictures or the P picture part of an Improved PB frame. (Possible filtering of B-pictures or the B-picture part of an Improved PB frame is not a matter for standardization, however some type of filtering is recommended for improved picture quality.) The capability of this mode is signaled by external means (for example Recommendation H.245). The use of this mode is indicated in the PLUSPTYPE field of the picture header.

J.2 Relation to UMV and AP modes (Annexes D and F)

The use of the deblocking filter mode has similar effects on picture quality as overlapped block motion compensation (OBMC) as defined in Annex F when used alone. When both techniques are used together, a further improvement in picture quality can be obtained. The Advanced Prediction mode (see also Annex F) consists of three elements:

1. Four motion vectors per macroblock as defined in section F.2
2. Overlapped motion compensation for luminance as defined in section F.3
3. Motion vectors over picture boundaries as defined in section D.1

In order for the Deblocking Filter mode to be able to provide maximum performance when complexity considerations may prevent use of the OBMC part of the Advanced Prediction mode, the Deblocking Filter mode includes the ability to use four motion vectors per macroblock and motion vectors over picture boundaries.

In summary, the three options defined in annexes D, F and J contain the following five coding elements:

1. Motion vectors over picture boundaries (D.1)
2. Extension of motion vector range (D.2)
3. Four motion vectors per macroblock (F.2)
4. Overlapped motion compensation for luminance (F.3)
5. Deblocking edge filter (J.3)

Table J.1 indicates which of the five elements are turned on depending on which of the three options defined in annexes D, F and J are turned on.

TABLE J.1/H.263

Feature Elements for UMV, AP, and DF modes

Unrestricted Motion Vector mode	Advanced prediction mode	Deblocking Filter mode	Motion vectors over picture boundaries	Extension of motion vector range	Four motion vectors per macroblock	Overlapped motion compensation for luminance	Deblocking edge filter
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
OFF	OFF	ON	ON	OFF	ON	OFF	ON
OFF	ON	OFF	ON	OFF	ON	ON	OFF
OFF	ON	ON	ON	OFF	ON	ON	ON
ON	OFF	OFF	ON	ON	OFF	OFF	OFF
ON	OFF	ON	ON	ON	ON	OFF	ON
ON	ON	OFF	ON	ON	ON	ON	OFF
ON	ON	ON	ON	ON	ON	ON	ON

J.3 Definition of the deblocking edge filter

The filter operations are performed across 8x8 block edges at the encoder as well as on the decoder side. The reconstructed image data (the sum of the prediction and reconstructed prediction error) are clipped to the range of 0 to 255 as described in section 6.3.2. Then the filtering is applied, which alters the picture that is to be stored in the picture store for future prediction. The filtering operations include an additional clipping to ensure that resulting pixel values stay in the range 0...255. No filtering is performed across a picture edge, and when the Independent Segment Decoding mode is in use, no filtering is performed across slice edges when the Slice Structured mode is in use (see Annexes K and R) or across the top boundary of GOBs having GOB headers present when the Slice Structured mode is not in use (see Annex R). Chrominance as well as luminance data are filtered.

When the mode described in this Annex is used together with the Improved PB-frames mode of Annex M, the backward prediction of the B-macroblock is based on the reconstructed P-macroblock (named P_{REC} in section G.5) after the clipping operation but before the deblocking edge filter operations. The forward prediction of the B-macroblock is based on the filtered version of the previous decoded picture (the same picture data that are used for prediction of the P-macroblock).

The deblocking filter operates using a set of four (clipped) pixel values on a horizontal or vertical line of the reconstructed picture, denoted as A, B, C and D, of which A and B belong to one block called block1 and C and D belong to a neighboring block called block2 which is to the right of or below block1. Figure J.1 shows examples for the position of these pixels.

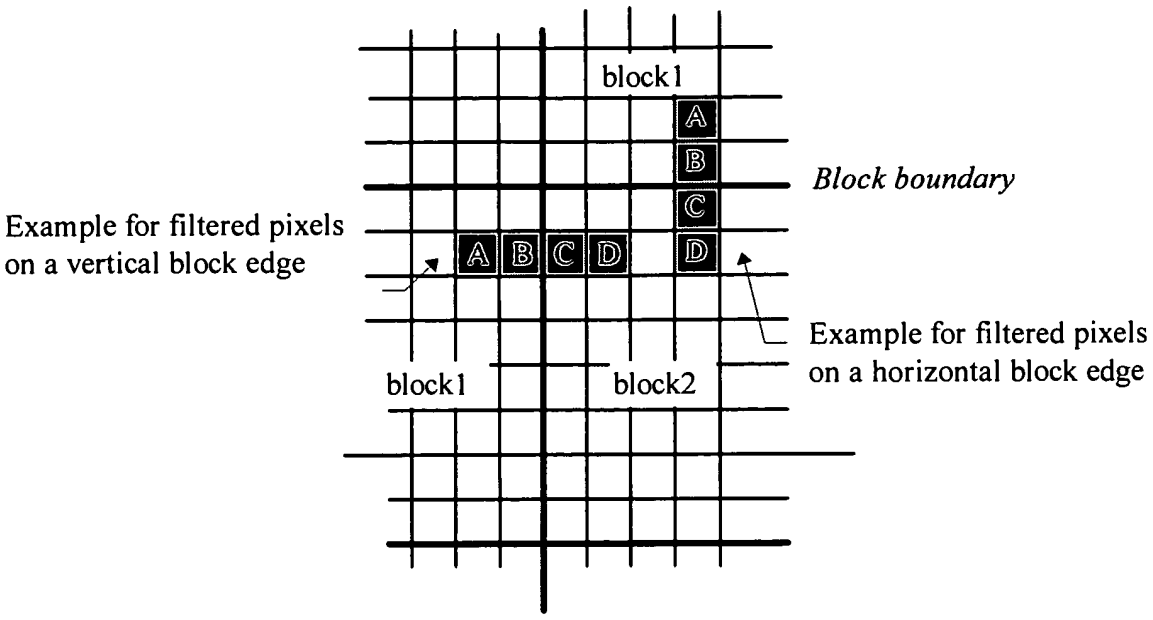


FIGURE J.1/H.263

Examples of positions of filtered pixels

One or both of the following conditions must be fulfilled in order to apply the filter across a particular edge:

Condition 1: block1 belongs to a coded macroblock (COD==0 || MB-type == INTRA), or

Condition 2: block2 belongs to a coded macroblock (COD==0 || MB-type == INTRA)

If filtering is to be applied across the edge, A, B, C, and D shall be replaced by A1, B1, C1, D1 where:

B1 = clip(B + d1)

C1 = clip(C – d1)

A1 = A – d2

D1 = D + d2

d = (A–4B+4C–D) / 8

d1 = UpDownRamp(d, STRENGTH)

d2 = clipd1((A – D) / 4, d1/2)

UpDownRamp(x, STRENGTH) = SIGN(x) * (MAX(0, abs(x)–MAX(0, 2*(abs(x) – STRENGTH))))

STRENGTH depends on QUANT and determines the amount of filtering.
The relation between STRENGTH and QUANT is given in Table J.2.

QUANT = quantization parameter used for block2 if block2 belongs to a coded macroblock, or

QUANT = quantization parameter used for block1 if block2 does not belong to a coded macroblock (but block1 does)

The function clip(x) is defined according to section 6.3.2, and the function clipd1(x, lim) clips x to the range ± abs(lim). The symbol "/" denotes division by truncation toward zero.

Figure J.2 shows how the value of d1 varies as a function of d. As a result, the filter has an effect only if d is smaller than 2*STRENGTH (and different from zero). This is to prevent the filtering of strong true edges in the

picture content. . However, if the Reduced-Resolution Update mode is in use, STRENGTH is set to infinity, and as a result, the value of d1 is always equal to the value of d (see section Q.7.2 in Annex Q).

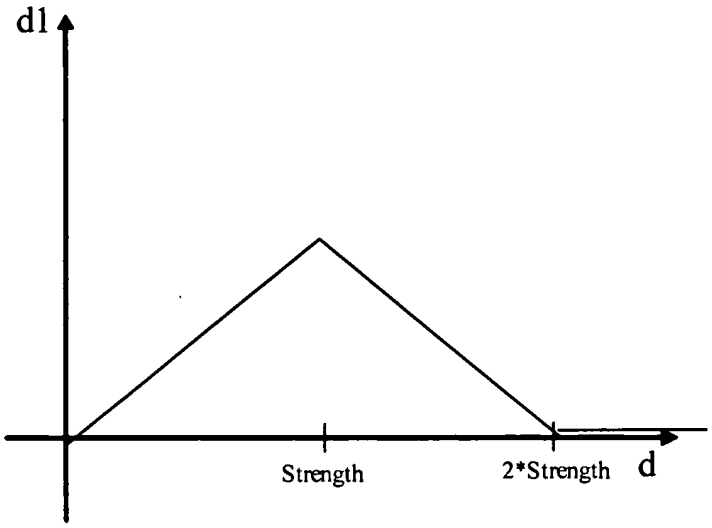


FIGURE J.2/H.263

Parameter d1 as a function of parameter d for deblocking filter mode

TABLE J.2/H.263

Relationship between QUANT and STRENGTH of Filter

QUANT	STRENGTH	QUANT	STRENGTH
1	1	17	8
2	1	18	8
3	2	19	8
4	2	20	9
5	3	21	9
6	3	22	9
7	4	23	10
8	4	24	10
9	4	25	10
10	5	26	11
11	5	27	11
12	6	28	11
13	6	29	12
14	7	30	12
15	7	31	12
16	7		

The definition of d1 is designed to ensure that small mismatches between the encoder and decoder will remain small and will not build up over multiple pictures of a video sequence. This would be a problem, for example, with a condition that simply switches the filter on or off, because a mismatch of only ±1 for d could then cause the filter to be switched on at the encoder side and off at the decoder side, or vice versa.

Due to rounding effects, the order of edges where filtering is performed must be specified.

Filtering across horizontal edges:

Basically this process is assumed to take place first. More precisely, the pixels $\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix}$ that are used in filtering across a horizontal edge shall not have been influenced by previous filtering across a vertical edge.

Filtering across vertical edges:

Before filtering across a vertical edge using pixels (A,B,C,D), all modifications of pixels (A,B,C,D) resulting from filtering across a horizontal edge shall have taken place.

Note that if one or more of the pixels (A,B,C,D) taking part in a filtering process are outside of a picture, no filtering takes place. Also, if the Independent Segment Decoding Mode is in use (see Annex R) and one or more of the pixels (A,B,C,D) taking part in a filtering process are in different video picture segments (see Annex I, section I.3 concerning when a block is considered to be in the same video picture segment), no filtering is performed.

Annex K
Slice Structured mode

(This annex forms an integral part of this Recommendation.)

K.1 Introduction

This annex describes the optional Slice Structured mode of H.263. The capability of this mode of H.263 is signaled by external means (for example Recommendation H.245). The use of this mode is indicated in the PLUSPTYPE field of the picture header. In order to facilitate optimal usage in a number of environments, this mode contains two submodes which are also capable of being signalled by external means (for example Recommendation H.245). These two submodes are used to indicate whether or not rectangular slices will be used and/or whether the slices will be transmitted in sequential order or sent in an arbitrary order.

A slice is defined as a slice header followed by consecutive macroblocks in scanning order. An exception is for the slice that immediately follows the picture start code in the bitstream for a picture (which is not necessarily the slice starting with macroblock 0). In this case only part of the slice header is transmitted as described in section K.2. The slice layer defines a video picture segment, and is used in place of the GOB layer in this optional mode. A slice video picture segment starts at a macroblock boundary in the picture and contains a number of macroblocks. Different slices within the same picture shall not overlap with each other, and every macroblock shall belong to one and only one slice.

This mode contains two submodes, which are signaled in the SSS field of the picture header:

1. The Rectangular Slice submode (RS). When RS is in use, the slice shall occupy a rectangular region of width specified by the SWI parameter of the slice header in units of macroblocks, and contains a number of macroblocks in scanning order within the rectangular region. When the Rectangular Slice submode is not in use, the SWI field is not present in the slice header and a slice contains a number of macroblocks in scanning order within the picture as a whole.
2. The Arbitrary Slice Ordering submode (ASO). When ASO is in use, the slices may appear in any order within the bitstream. When ASO is not in use, the slices must be sent in the (unique) order for which the MBA field of the slice header is strictly increasing from each slice to each subsequent slice in the picture.

Slice boundaries are treated differently than simple macroblock boundaries in order to allow slice header locations within the bitstream to act as resynchronization points for bit error and packet loss recovery and in order to allow out-of-order slice decoding within a picture. Thus, no data dependencies can cross the slice boundaries within the current picture, except for the Deblocking Filter mode which, when in use without the Independent Segment Decoding mode, filters across the boundaries of the blocks in the picture. However, motion vectors within a slice can cause data dependencies which cross the slice boundaries in the reference picture used for prediction purposes, unless the optional Independent Segment Decoding mode is in use.

The following rules are adopted to ensure that slice boundary locations can act as resynchronization points and to ensure that slices can be sent out of order without causing additional decoding delays:

1. The prediction of motion vector values are the same as if a GOB header were present (see section 6.1.1), preventing the use of motion vectors of blocks outside the current slice for the prediction of the values of motion vectors within the slice.
2. The Advanced INTRA Coding mode (see Annex I) treats the slice boundary as if it were a picture boundary with respect to the prediction of INTRA block DCT coefficient values.
3. The assignment of remote motion vectors for use in overlapped block motion compensation within the Advanced Prediction mode also prevents the use of motion vectors of blocks outside the current slice for use as remote motion vectors (see section F.3).

K.2 Structure of slice layer

The structure of the slice layer of the syntax is shown in Figure K.1 for all slices except the slice that immediately follows the picture start code in the bitstream for a picture. For the slice following the picture start code, only the emulation prevention bits (SEPB1, SEPB3, and conditionally SEPB2 as specified below), the MBA field and, when in the RS submode, also the SWI field are included.

SSTUF	SSC	SEPB1	SSBI	MBA	SEPB2	SQUANT	SWI	SEPB3	GFID	Macroblock Data

FIGURE K.1/H.263

Structure of slice layer

Refer to section 5.2.5 for GFID, and section 5.3 for description of Macroblock layer.

K.2.1 Stuffing (SSTUF) (Variable Length)

A codeword of variable length consisting of less than 8 bits. Encoders shall insert this codeword directly before an SSC codeword whenever needed to ensure that SSC is byte-aligned. If SSTUF is present, the last bit of SSTUF shall be the last (least significant) bit of a byte, so that the start of the SSC codeword is byte aligned. Decoders shall be designed to discard SSTUF. Notice that 0 is used for stuffing within SSTUF.

K.2.2 Slice Start Code (SSC) (17 bits)

A word of 17 bits. Its value is 0000 0000 0000 0000 1. Slice start codes shall be byte aligned. This can be achieved by inserting SSTUF before the start code such that the first bit of the start code is the first (most significant) bit of a byte. The slice start code is not present for the slice which follows the picture start code.

K.2.3 Slice Emulation Prevention Bit 1 (SEPB1) (1 bit)

A single bit always having the value of "1", which is included to prevent start code emulation.

K.2.4 Slice Sub-Bitstream Indicator (SSBI) (4 bits)

A codeword of length four bits which is present only when CPM = '1' in the picture header. SSBI indicates the Sub-Bitstream number for the slice for Continuous Presence Multipoint and Video Multiplex operation, as described in Annex C. The mapping from the value of SSBI to the Sub-Bitstream number is shown in Table K.1. SSBI is not present for the slice which follows the picture start code.

K.2.5 Macroblock Address (MBA) (5/6/7/9/11/12/13/14 bits)

A codeword having a length that is dependent on the current picture size and whether the Reduced Resolution Update mode is in effect (see Annex Q). The bits are the binary representation of the macroblock number of the first macroblock in the current slice as counted from the beginning of the picture in scanning order, starting with macroblock number 0 at the upper left corner. MBA uniquely identifies which macroblock in the picture the current slice starts with. Codeword lengths for this codeword are provided in Table K.2. For custom picture sizes, the field width is given by the first entry in the table that has an equal or larger number of macroblocks, and the maximum value is the number of macroblocks in the current picture minus one. When in the Reduced Resolution Update mode, the relevant picture size is the lower resolution update picture size rather than the picture size indicated in the picture header (see Annex Q).

K.2.6 Slice Emulation Prevention Bit 2 (SEPB2) (1 bit)

A single bit always having the value "1", which is included under certain conditions to prevent start code emulation. For slices other than the one following the picture start code, SEPB2 is included only if the MBA field width is greater than 11 bits and CPM = '0' in the picture header, or if the MBA field width is greater than 9 bits and CPM = '1' in the picture header. For the slice following the picture start code, SEPB2 is included only if the Rectangular Slice submode is in use.

K.2.7 Quantizer Information (SQUANT) (5 bits)

A fixed length codeword of 5 bits which indicates the quantizer QUANT to be used for that slice until updated by any subsequent DQUANT. The codewords are the natural binary representations of the values of QUANT which, being half the step sizes, range from 1 to 31. SQUANT is not present for the slice which follows the picture start code.

K.2.8 Slice Width Indication in Macroblocks (SWI) (3/4/5/6/7 bits)

A codeword which is present only if the Rectangular Slice submode is active, and having a length which depends on the current picture size and whether the Reduced Resolution Update mode is active, as specified in Table K.3. For custom picture sizes, the field width is given by the next standard format size which is equal or larger in width (QCIF, CIF, ...), and the maximum value is the total number of macroblocks across the picture minus one. The last row of the table indicates the field width for picture sizes wider than 16CIF. SWI refers to the width of the current

rectangular slice having its first macroblock (upper left) specified by MBA. The calculation of the actual slice width is given by

Actual Slice Width = SWI + 1

When in the Reduced Resolution Update mode, the relevant picture size is that of the lower resolution picture size for the update information, rather than the picture size indicated in the picture header.

K.2.9 Slice Emulation Prevention Bit 3 (SEPB3) (1 bit)

A single bit always having the value of "1" in order to prevent start code emulation.

TABLE K.1/H.263

Values of SSBI and associated Sub-Bitstream Numbers

Sub-Bitstream Number	SSBI Field Value	GN Value Emulated
0	1001	25
1	1010	26
2	1011	27
3	1101	29

TABLE K.2/H.263

Specification of MBA parameter

	Default		RRU mode	
Picture Format	Max Value	Field Width	Max Value	Field Width
sub-QCIF	47	6	11	5
QCIF	98	7	29	6
CIF	395	9	98	7
4CIF	1583	11	395	9
16CIF	6335	13	1583	11
2048x1152	9215	14	2303	12

TABLE K.3/H.263

Specification of SWI parameter

	Default		RRU mode	
Picture Format	Max Value	Field Width	Max Value	Field Width
sub-QCIF	7	4	3	3
QCIF	10	4	5	3
CIF	21	5	10	4
4CIF	43	6	21	5
16CIF	87	7	43	6
1412...2048 pixels wide	127	7	63	6

Annex L
Supplemental Enhancement Information Specification

(This annex forms an integral part of this Recommendation.)

L.1 Introduction

This annex describes the format of the supplemental enhancement information sent in the PSUPP field of the picture layer of H.263. The capability of a decoder to provide any or all of the enhanced capabilities described in this annex may be signaled by external means (for example Recommendation H.245). Decoders which do not provide the enhanced capabilities may simply discard any PSUPP information bits that appear in the bitstream. The presence of this supplemental enhancement information is indicated in PEI, and an additional PEI bit is inserted between every octet of PSUPP data, as described in sections 5.1.24 and 5.1.25.

In this annex, a distinction is made between the "decoded picture" and the "displayed picture". For purposes of this annex, the "displayed picture" is a picture having the same picture format as specified for the current picture by the picture layer of the video bitstream syntax. The "displayed picture" is constructed as described in this annex from the decoded picture, the prior displayed picture, the supplementary enhancement information described herein, and in some cases partly from a background picture which is externally controlled.

L.2 PSUPP Format

The PSUPP data consists of a four-bit function type indication FTYPE, followed by a four-bit parameter data size specification DSIZE, followed by DSIZE octets of function parameter data, optionally followed by another function type indication, and so on. One function type indication value is defined as an escape code to provide for future extensibility to allow definition of more than fifteen different functions. A decoder which receives a function type indication which it does not support can discard the function parameter data for that function and then check for a subsequent function type indication which may be supported. The defined FTYPE values are shown in Table L.1.

TABLE L.1/H.263

FTYPE Function Type Values

0	Reserved
1	Do Nothing
2	Full-Picture Freeze Request
3	Partial-Picture Freeze Request
4	Resizing Partial-Picture Freeze Request
5	Partial-Picture Freeze-Release Request
6	Full-Picture Snapshot Tag
7	Partial-Picture Snapshot Tag
8	Video Time Segment Start Tag
9	Video Time Segment End Tag
10	Progressive Refinement Segment Start Tag
11	Progressive Refinement Segment End Tag
12	Chroma Key Information
13	Reserved
14	Reserved
15	Extended Function Type

L.3 Do Nothing

No action is requested by the Do Nothing function. This function is used to prevent start code emulation. Whenever the last five or more bits of the final octet of the previous PSUPP octet are all zero and no additional PSUPP function requests are to be sent, the Do Nothing function shall be inserted into PSUPP to prevent the possibility of start code emulation. The Do Nothing function may also be sent when it is not required by rule expressed in the previous sentence. DSIZE shall be zero for the Do Nothing function.

L.4 Full-Picture Freeze Request

The full-picture freeze request function indicates that the contents of the entire prior displayed video picture shall be kept unchanged, without updating the displayed picture using the contents of the current decoded picture. The displayed picture shall then remain unchanged until the freeze picture release bit in the current PTYPE or in a subsequent PTYPE is set to 1, or until timeout occurs, whichever comes first. The request shall lapse due to timeout after five seconds or five pictures, whichever is a longer period of time. The timeout can be prevented by the issuance of another full-picture freeze request prior to or upon expiration of the timeout period (e.g., repeating the request in the header of the first picture with temporal reference indicating a time interval greater than or equal to five seconds since issuance, or in the header of the fifth picture after issuance). DSIZE shall be zero for the full-picture freeze request function.

L.5 Partial-Picture Freeze Request

The partial-picture freeze request function indicates that the contents of a specified rectangular area of the prior displayed video picture should be kept unchanged, without updating the specified area of the displayed picture using the contents of the current decoded picture. The specified area of the displayed picture shall then remain unchanged until the freeze picture release bit in the current PTYPE or in a subsequent PTYPE is set to 1, until a partial-picture freeze-release request affecting the specified area is received, until the source format specified in a picture header differs from that of previous picture headers, or until timeout occurs, whichever comes first. Any change in the picture source format shall act as a freeze release for all active partial-picture freeze requests. The request shall lapse due to timeout after five seconds or five pictures, whichever is a longer period of time. The timeout can be prevented by the issuance of an identical partial-picture freeze request prior to or upon expiration of the timeout period (e.g., repeating the request in the header of the first picture with temporal reference indicating a time interval greater than or equal to five seconds since issuance, or in the header of the fifth picture after issuance). DSIZE shall be equal to 4 for the partial-picture freeze request. The four octets of PSUPP that follow contain the horizontal and vertical location of the upper left corner of the frozen picture rectangle, and the width and height of the rectangle, respectively, using eight bits each and expressed in units of eight pixels. For example, a 24-pixel wide and 16 pixel tall area in the upper left corner of the video display is specified by the four parameters (0, 0, 3, 2).

L.6 Resizing Partial-Picture Freeze Request

The resizing partial-picture freeze request function indicates that the contents of a specified rectangular area of the prior displayed video picture should be resized to fit into a smaller part of the displayed video picture, which should then be kept unchanged, without updating the specified area of the displayed picture using the contents of the current decoded picture. The specified area of the displayed picture shall then remain unchanged until the freeze release bit in the current PTYPE or in a subsequent PTYPE is set to 1, until a partial-picture freeze-release request affecting the specified area is received, until the source format specified in a picture header differs from that of previous picture headers, or until timeout occurs, whichever comes first. Any change in the picture source format shall act as a freeze release for all active resizing partial-picture freeze requests. The request shall lapse due to timeout after five seconds or five pictures, whichever is a longer period of time. The timeout can be prevented by the issuance of a partial-picture freeze request for the affected area of the displayed picture prior to or upon expiration of the timeout period (e.g., issuing a partial-picture freeze request in the header of the first picture with temporal reference indicating a time interval greater than or equal to five seconds since issuance, or in the header of the fifth picture after issuance). DSIZE shall be equal to 8 for the resizing partial-picture freeze request. The eight octets of PSUPP data that follow contain 32 bits used to specify the rectangular region of the affected area of the displayed picture, and then 32 bits used to specify the corresponding rectangular region of the affected area of the decoded picture. The width and height of the rectangular region in the decoded picture shall both be equal to 2^i times the width and height specified for the rectangular region in the displayed picture, where i is an integer in the range of 1 to 8. The location and size of each of these two rectangular regions is specified using the same format as such a region is specified in the partial-picture freeze request function.

L.7 Partial-Picture Freeze-Release Request

The partial-picture freeze-release request function indicates that the contents of a specified rectangular area of the displayed video picture shall be updated by the current and subsequent decoded pictures. DSIZE shall be equal to 4 for the partial-picture freeze-release request. The four octets of PSUPP data that follow specify a rectangular region of the displayed picture in the same format as such a region is specified in the partial-picture freeze request function.

L.8 Full-Picture Snapshot Tag

The full-picture snapshot tag function indicates that the current picture is labeled for external use as a still-image snapshot of the video content. DSIZE shall be equal to 4 for the full-picture snapshot tag function. The four octets of PSUPP data that follow specify a snapshot identification number for external use.

L.9 Partial-Picture Snapshot Tag

The partial-picture snapshot tag function indicates that a specified rectangular area of the current picture is labeled for external use as a still-image snapshot of the video content. DSIZE shall be equal to 8 for the partial-picture snapshot tag function. The first four octets of PSUPP data that follow specify a snapshot identification number for external use, and the remaining four octets of PSUPP data that follow specify a rectangular region of the decoded picture in the same format as such a region is specified in the partial-picture freeze request function.

L.10 Video Time Segment Start Tag

The video time segment start tag function indicates that the beginning of a specified sub-sequence of video data is labeled as a useful section of video content for external use, starting with the current picture. The tagged sub-sequence of video data shall continue until stopped by the receipt of a matching video time segment end tag function or until timeout, whichever comes first. The tagged sub-sequence shall end due to timeout after five seconds or five pictures, whichever is a longer period of time. The timeout can be prevented by the issuance of an identical video time segment start tag function prior to or upon expiration of the timeout period (e.g., repeating the video time segment start tag function in the header of the first picture with temporal reference indicating a time interval greater than or equal to five seconds since issuance, or in the header of the fifth picture after issuance). DSIZE shall be equal to 4 for the video time segment start tag function. The four octets of PSUPP data that follow specify a video time segment identification number for external use.

L.11 Video Time Segment End Tag

The video time segment end tag function indicates that the end of a specified sub-sequence of video data is labeled as a useful section of video content for external use, ending with the previous picture. DSIZE shall be equal to 4 for the video time segment start tag function. The four octets of PSUPP data that follow specify a video time segment identification number for external use.

L.12 Progressive Refinement Segment Start Tag

The progressive refinement segment start tag function indicates the beginning of a specified sub-sequence of video data which is labeled as the current picture followed by a sequence of zero or more pictures of refinement of the quality of the current picture, rather than as a representation of a continually moving scene. The tagged sub-sequence of video data shall continue until stopped by the receipt of a matching progressive refinement segment end tag function or until timeout, whichever comes first. The tagged sub-sequence shall end due to timeout after five seconds or five pictures, whichever is a longer period of time. The timeout can be prevented by the issuance of an identical progressive refinement segment start tag function prior to or upon expiration of the timeout period (e.g., repeating the progressive refinement start tag function in the header of the first picture with temporal reference indicating a time interval greater than or equal to five seconds since issuance, or in the header of the fifth picture after issuance). DSIZE shall be equal to 4 for the video time segment start tag function. The four octets of PSUPP data that follow specify a progressive refinement segment identification number for external use.

L.13 Progressive Refinement Segment End Tag

The progressive refinement segment end tag function indicates the end of a specified sub-sequence of video data which is labeled as an initial picture followed by a sequence of zero or more pictures of the refinement of the quality of the initial picture, and ending with the previous picture. DSIZE shall be equal to 4 for the video time segment start tag function. The four octets of PSUPP data that follow specify a progressive refinement identification number for external use.

L.14 Chroma Keying Information

The Chroma Keying Information Function (CKIF) indicates that the "chroma keying" technique is used to represent "transparent" and "semi-transparent" pixels in the decoded video pictures. When being presented on the display, "transparent" pixels are not displayed. Instead, a background picture which is either a prior reference picture or is an externally controlled picture is revealed. Semi-transparent pixels are displayed by blending the pixel value in

the current picture with the corresponding value in the background picture. One octet is used to indicate the keying color value for each component (Y, C_B, or C_R) which is used for chroma keying. To represent pixels that are to be "semi-transparent," two threshold values, denoted as T_1 and T_2, are used. Let α denote the transparency of a pixel; $\alpha = 255$ indicates that the pixel is opaque, and $\alpha = 0$ indicates that the pixel is transparent. For other values of α, the resulting value for a pixel should be a weighted combination of the pixel value in the current picture and the pixel value from the background picture (which is specified externally). The values of α may be used to form an image that is called an "alpha map." Thus the resulting value for each component may be

$$[\alpha \cdot X + (255-\alpha) \cdot Z] / 255$$

where X is the decoded pixel component value (for Y, C_B, or C_R), and Z is the corresponding pixel component value from the background picture.

The α value can be calculated as follows. First, the distance of the pixel color from the key color value is calculated:

$$d = A_Y(X_Y - K_Y)^2 + A_B(X_B - K_B)^2 + A_R(X_R - K_R)^2$$

in which X_Y, X_B, and X_R are the Y, C_B, and C_R values of the decoded pixel color, K_Y, K_B, and K_R are the corresponding key color parameters, and A_Y, A_B, and A_R. are keying flag bits which indicate which color components are used as keys. Once the distance d is calculated, the α value may be computed as specified in the following pseudo-code:

```
for each pixel
   if      (d<T1) then α = 0;
   else if (d>T2) then α = 255;
   else α = [255·(d-T1)]/(T2-T1)
```

However, the precise method for performing the chroma keying operation in the decoder is not specified herein, since normative specification of the method is not needed for interoperability. The process described here is provided for illustration purposes in order to convey the intended interpretation of the data parameters.

Since the derived α value is simply a function of X_Y, X_B, and X_R, a look-up table (LUT) can be built to achieve the above operation. Such an LUT has 2^{8xN} entries corresponding to all pixel values, where N is the number of color components used as keys. Each entry in the LUT would then contain the corresponding α value.

DSIZE shall be in the range of 1 to 9 (inclusive) for chroma keying information, according to the amount of data sent with the CKIF. No more than one CKIF shall be sent with a picture.

The first octet following the DSIZE octet shall contain the representation order of the current picture—streams having a lower representation order are assumed to form the background picture for streams having a higher representation order.

If DSIZE is greater than one, the next octet after the representation order octet shall be used to send six flag bits defined as:

bit 1 : A_Y : A flag bit indicating the presence of a K_Y key parameter for luminance Y values
bit 2 : A_B : A flag bit indicating the presence of a K_B key parameter for chroma C_B values
bit 3 : A_R : A flag bit indicating the presence of a K_R key parameter for chroma C_R values
bit 4 : A_1 : A flag bit indicating the presence of a T_1 threshold parameter for transparency
bit 5 : A_2 : A flag bit indicating the presence of a T_2 threshold parameter for opacity
bit 6 : RPB: A flag bit indicating the use of the reference picture as a background picture
bit 7 : reserved
bit 8 : reserved

DSIZE shall be equal to 1 or shall be equal to 2 plus the number of flag bits among A_Y, A_B, and A_R which are set to 1, plus 2 times the number of bits among A_1 and A_2 which are set to 1. If DSIZE is greater than 1, then an additional octet shall be sent to specify the value of each color component for each of the flag bits A_Y, A_B, and A_R which are set to one and two additional octets shall be sent to specify each of the flagged threshold values among T_1 and T_2. These octets shall follow in the same order as the flag bits.

If DSIZE is equal to 1, or if all three keying color flag bits A_Y, A_B, and A_R, are zero, the keying flag bits A_Y, A_B, and A_R and keying colors K_Y, K_B, and K_R which were used for the previous keyed picture should also be used for the current picture. If no previous values have been sent for the video sequence, the default keying flag bits $A_Y = 1$, $A_B = 1$, and $A_R = 1$ and the default key colors $K_Y = 50$, $K_B = 220$, and $K_R = 100$ should be used as the previous values.

If DSIZE is equal to 1, or if both of the keying threshold flag bits A_1 and A_2 are zero, the keying threshold values T_1 and T_2 which were used for the previous keyed picture should also be used for the current picture. If no previous values have been sent for the video sequence, the default threshold values $T_1 = 48$ and $T_2 = 75$ should be used as the previous values.

In the portion where the pixels are "semi-transparent" (i.e., when $T_1 < d < T_2$), the decoded pixels typically contain the chroma key color in components where chroma keys are used. This may result in certain color artifacts. To address this problem, these pixel values may be adjusted before they are blended with the background color. Such a correction process may be applied to color components being used in the chroma key operation as indicated by the flag bits. The process is as follows:

$$X' = K + (T_2/d)\,(X-K)$$

where X is the original decoded pixel component value, and X' is the corrected value.

Since the adusted pixel values X'_Y, X'_B, and X'_R are functions of X_Y, X_B, and X_R, the color correction can be achieved by using an LUT. This LUT would have 2^{8N} entries corresponding to all pixel values, where N is the number of color components used as keys. Each entry would then contain the corresponding corrected values.

If the reference picture background (RPB) flag bit is set to '1', this indicates that the temporally-previous reference picture (prior to any Annex P resampling performed for the current picture) should be held as the (opaque) background of the current picture and of all subsequent chroma keyed pictures, until replaced by the arrival of another picture having an RPB flag set to '1'. If the current picture has no temporally-previous reference picture (i.e., if the current picture is a picture of type INTRA or EI), the picture to which the RPB flag bit refers is the picture which would normally have been the reference picture if the current picture were of type INTER or EP as appropriate. If the RPB flag bit is set to '0', this indicates that background should remain as previously controlled (either under external control or using a reference picture which was previously stored upon receipt of a prior picture having RPB set to '1').

The use of the chroma keying which is invoked by the issuance of the chroma keying information function shall start with the current picture and shall continue until a subsequent picture of type INTRA or EI occurs or until a timeout period expires, whichever comes first. The use of chroma keying shall end due to timeout after five seconds or five pictures, whichever is a longer period of time. The timeout can be prevented by the issuance of an identical chroma keying information function prior to or upon expiration of the timeout period (e.g., repeating the chroma keying information function in the header of the first picture with temporal reference indicating a time interval greater than or equal to five seconds since issuance, or in the header of the fifth picture after issuance). The encoder shall send sufficient information with the chroma keying information function for complete resynchronization to occur with each picture of type INTRA or EI and within each timeout interval (it shall not rely on using stored or default values of the key colors or thresholds).

L.15 Extended Function Type

The extended function type indication is used to signal that the following PSUPP octet contains an extended function. The usage of extended functions is reserved for the ITU to have a later ability to define a larger number of backward-compatible PSUPP data functions. DSIZE shall be equal to zero for the extended function type indication. In order to allow backward compatibility of future use of the extended function type indication, decoders shall treat the second set of four bits in the octet which follows the extended function type indication as a DSIZE value indicating the number of subsequent octets of PSUPP that are to be skipped for extended function parameter data, which may be followed by additional FTYPE indications.

Annex M
Improved PB-frames mode

(This annex forms an integral part of this Recommendation.)

M.1 Introduction

This annex describes an optional Improved PB-frames mode of H.263. It is therefore considered to be advantageous to use the present Improved PB-frames mode instead of the PB-frames mode defined in Annex G. The capability of this mode is signaled by external means (for example Recommendation H.245). The use of this mode is indicated in the PLUSPTYPE field of the picture header.

Most parts of this option are similar to the PB-frame option defined in Annex G.

To avoid confusion with B pictures as defined in Annex O, the terms B-picture, B-macroblock, and B-block will not be used in this annex. Instead, we will use the notation B_{PB} to mean the "B-part" of an Improved PB frame. When reference is made to Annex G, B-picture and B-block shall be read as B_{PB}-picture and B_{PB}-block.

The main difference between the PB frames mode and the Improved PB frames mode is that in the Improved PB frames mode, the B_{PB}-macroblock has available a forward and a backward prediction mode in addition to the bi-directional prediction mode. In this Annex, MVDB (when present) refers to a forward motion vector. (Note that in Annex G MVDB was used to refer to an enhancement of the downscaled forward and backward vectors for bidirectional prediction, rather than a distinct forward motion vector.)

All the differences from Annex G are identified in this annex. When nothing is indicated, it means that the same procedure as described in Annex G is used.

M.2 B_{PB}-macroblock prediction modes

There are three different ways of coding a B_{PB}-macroblock. The different coding modes are signaled by the parameter MODB. The B_{PB}-macroblock coding modes are:

M.2.1 Bidirectional prediction

In the bidirectional prediction mode, prediction uses the reference pictures before and after the B_{PB}-picture (in the case of a sequence of Improved PB frames, this means the P-picture part of the temporally previous Improved PB frame and the P-picture part of the current Improved PB frame). This prediction is equivalent to the prediction defined in Annex G when MV_D =0. Notice that in this mode (and only in this mode), motion vector data (MVD) of the PB-macroblock must be included even if the P-macroblock is INTRA coded. (Notice the difference between MVD - motion vector data - and MV_D - delta vector - defined in Annex G).

M.2.2 Forward prediction

In the forward prediction mode the vector data contained in MVDB are used for forward prediction from the previous reference picture (an INTRA or INTER picture, or the P-picture part of a PB or Improved PB frame). This means that there is always only one 16x16 vector for the B_{PB}-macroblock in this prediction mode.

A simple predictor is used for coding of the forward motion vector. The rule for this predictor is that if the current macroblock is not at the far left edge of the picture or slice and the macroblock to the left has a forward motion vector, then the predictor of the forward motion vector for the current macroblock is set to the value of the forward motion vector of the block to the left; otherwise the predictor is set to zero. The difference between the predictor and the desired motion vector is then VLC coded in the same way as vector data to be used for the P-picture (MVD).

Concerning motion vectors over picture boundaries defined in section D.1, the described technique also applies for the forward B_{PB}-vector if this feature is in use (this applies for forward as well as for bidirectional prediction mode).

M.2.3 Backward prediction

In the backward prediction mode the prediction of the B_{PB} macroblock is identical to P_{REC} (defined in section G.5). No motion vector data is used for the backward prediction.

M.3 Calculation of vectors for bidirectional prediction of a the B-macroblock

In case bidirectional prediction is used, the scaled forward and backward vectors are calculated as described in Annex G when $MV_D = 0$.

M.4 MODB table

A new definition for MODB (replacing Table 11) is shown in Table M.1. It indicates the possible coding modes for a B_{PB}-block.

TABLE M.1/H.263

MODB table for Improved PB-frames mode

Index	CBPB	MVDB	Number of bits	Code	Coding Mode
0			1	0	Bidirectional prediction
1	x		2	10	Bidirectional prediction
2		x	3	110	Forward prediction
3	x	x	4	1110	Forward prediction
4			5	11110	Backward prediction
5	x		5	11111	Backward prediction

Note — The symbol "x" in the table above indicates that the associated syntax element is present.

Annex N
Reference Picture Selection mode

(This annex forms an integral part of this Recommendation.)

N.1 Introduction

This annex describes the optional Reference Picture Selection mode of H.263, which operates using a modified interframe prediction method called "NEWPRED." The capability of this mode of H.263 is signaled by external means (for example Recommendation H.245). The amount of additional picture memory accommodated in the decoder may also be signaled by external means to help the memory management at the encoder. This mode can use backward channel messages sent from a decoder to an encoder to inform the encoder which part of which pictures have been correctly decoded at the decoder. The use of this mode is indicated in the PLUSPTYPE field of the picture header. This mode has two back-channel mode switches which define whether a backward channel is used and what kind of messages are returned on that backward channel from the decoder, and has another submode defined in terms of the channel for the backward channel messages.

The two back-channel mode switches of this mode determine the type of messages sent on the back-channel, specifying whether ACK (acknowledgment messages), or NACK (non-acknowledgment messages) are sent. Together, the two switches define four basic methods of operation:

1. NEITHER: In which no back-channel data is returned from the decoder to the encoder,
2. ACK: In which the decoder returns only acknowledgment messages,
3. NACK: In which the decoder returns only non-acknowledgment messages, and
4. ACK+NACK: In which the decoder returns both acknowledgment and non-acknowledgment messages.

The specific type of messages to be sent as outlined above is indicated in the picture header.

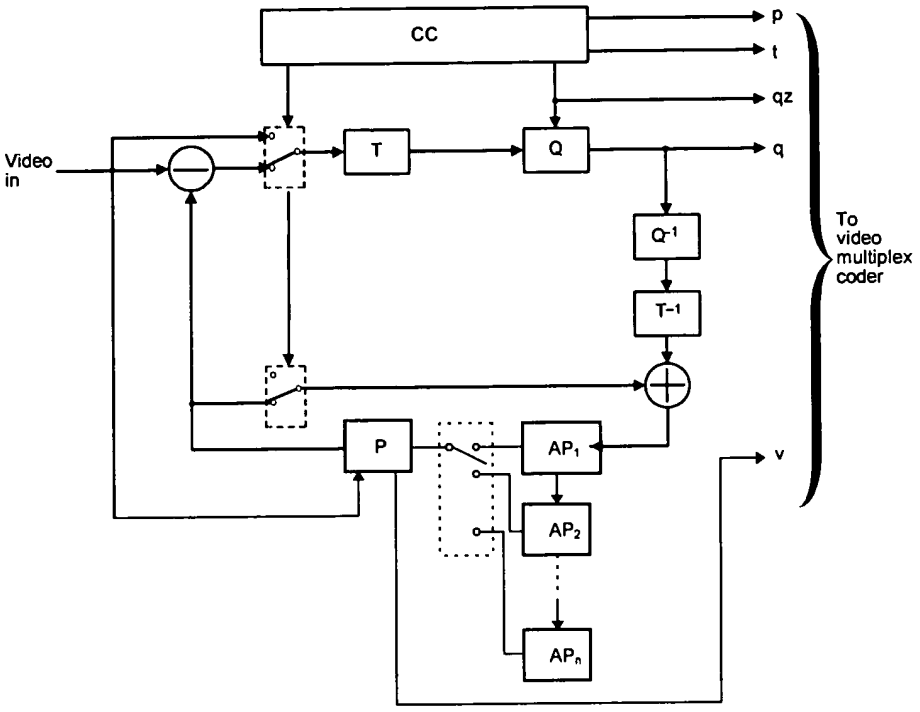
There are also two methods of operation in terms of the channel for backward channel messages:

1. Separate Logical Channel mode: This method of operation delivers back-channel data through a separate logical channel in the multiplex layer of the system, and
2. VideoMux mode: This method of operation delivers back-channel data for received video within the forward video data of a video stream of encoded data.

This annex specifies a syntax for the backward channel messages as well as for forward channel data.

N.2 Video source coding algorithm

The source coder of this mode is shown in generalized form in Figure N.1. This figure shows a structure which uses a number of picture memories. The source coder may select one of the picture memories to suppress the temporal error propagation due to the inter-frame coding. The Independent Segment Decoding mode (see Annex R), which treats boundaries of GOBs with non-empty headers or slices as picture boundaries, can be used to avoid error propagation due to motion compensation across the boundaries of the GOBs or slices when this mode is applied to a smaller unit than a picture, such as a GOB or slice. The information to signal which picture is selected for prediction is included in the encoded bitstream. The strategy used by the encoder to select the picture to be used for prediction is out of the scope of this Recommendation.



T Transform
Q Quantizer
P Picture Memory with motion compensated variable delay
AP Additional Picture Memory
CC Coding control
p Flag for INTRA/INTER
t Flag for transmitted or not
qz Quantizer indication
q Quantizing index for transform coefficients
v Motion vector

FIGURE N.1/H.263

Source coder for NEWPRED

N.3 Channel for Back-Channel messages

This mode has two methods of operation in terms of the type of channel for back-channel messages. One is the separate logical channel mode and the other is the videomux mode. The separate logical channel mode is a preferred mode and delivers the back-channel message defined in N.4.2 through the dedicated logical channel. The videomux mode is prepared for the system which cannot set up the separate extra channel for the back-channel messages due to the restriction of the number of channels' combinations. The videomux mode delivers the back-channel messages through the same logical channel with the forward video data in the opposite direction.

N.3.1 Separate Logical Channel Mode

The separate logical channel mode delivers back-channel messages through a dedicated logical channel opened only for the purpose of the back-channel messages. The association mechanism with the forward channel which delivers video data is provided by external means (for example Recommendation H.245). Separate logical channel operation requires an external framing mechanism for synchronization of the messages within the back channel, as the back-channel syntax defined herein contains no synchronization flag words.

N.3.2 VideoMux Mode

The videomux mode delivers the back-channel messages through the same logical channel with the forward video data in the opposite direction. The syntax of the multiplexed bitstream is described in the section N.4.1. The back-

channel messages may be inserted by using the Back-Channel message Indication (BCI) in the GOB or slice header.

N.4 Syntax

N.4.1 Forward Channel

The syntax for the forward channel data which conveys the compressed video signal is altered only in the Group of Blocks (GOB) or slice layer.

The syntax of the GOB layer is illustrated in Figure N.2. The fields of TRI, TR, TRPI, TRP, BCI, and BCM are added to Figure 9.

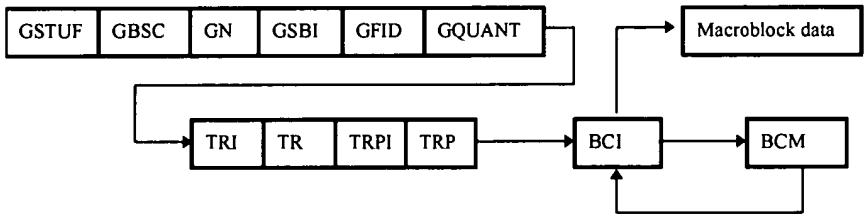


FIGURE N.2/H.263

Structure of GOB layer for NEWPRED

When the optional Slice Structured mode (see Annex K) is in use, the syntax of the slice layer is modified in the same way as the GOB layer. The syntax is illustrated Figure N.3.

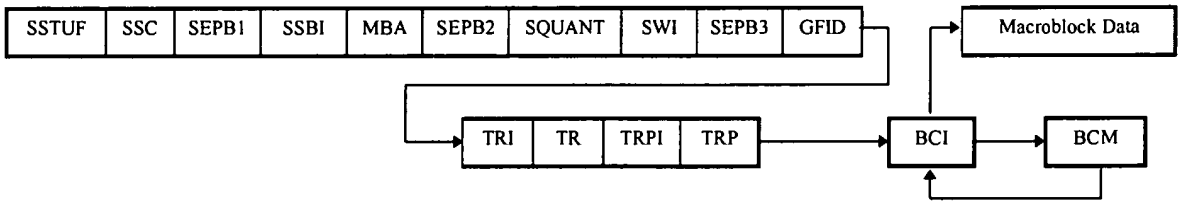


FIGURE N.3/H.263

Structure of Slice layer for NEWPRED

N.4.1.1 TRI (Temporal Reference Indicator) (1 bit)

TRI indicates whether or not the following TR field is present.

0: TR field is not present.
1: TR field is present.

N.4.1.2 Temporal Reference (TR) (8/10 bits)

When present, TR is an eight bit number unless a custom picture clock frequency is in use, in which case it is a ten bit number consisting of the concatenation of ETR and TR of the picture header.

N.4.1.3 TRPI (Temporal Reference for Prediction Indicator) (1 bit)

TRPI indicates whether or not the following TRP field is present.

0: TRP field is not present.
1: TRP field is present.

TRPI shall be equal to zero whenever the picture is an I or EI picture.

N.4.1.4 Temporal Reference for Prediction (TRP) (10 bits)

When present (as indicated in TRPI), TRP indicates the Temporal Reference which is used for prediction of the encoding, except in the case of B pictures and the B-picture part of an Improved PB frame. For B pictures or the B-picture part of an Improved PB frame, the picture having the temporal reference TRP is used for the prediction in the forward direction. (Prediction in the reverse-temporal direction always uses the immediately temporally-subsequent picture.) TRP is a ten bit number. If a custom picture clock frequency was not in use for the reference picture, the two MSBs of TR are zero and the LSBs contain the eight bit TR found in the picture header of the reference picture. If a custom picture clock frequency was in use for the reference picture, TR is a ten bit number consisting of the concatenation of ETR and TR from the reference picture header.

N.4.1.5 Back-Channel message Indication (BCI) (Variable Length)

This field contains one or two bits, when set to "1", it signals the presence of the following video Back-Channel Message (BCM) field. Otherwise, the field value is "01", which indicates the absence or the end of the video back-channel message field. Combinations of BCM and BCI may not be present, and may be repeated when present. BCI shall always be set to "01" if the videomux mode is not in use.

N.4.1.6 Back-Channel Message (BCM) (Variable Length)

The Back-Channel Message having syntax as defined in the section N.4.2, which is present only if the preceding BCI field is set to "1".

N.4.2 Back-Channel Message (BCM) Syntax

The syntax for the back-channel which conveys the acknowledgment/non-acknowledgment messages is illustrated in Figure N.4. This message is returned from a decoder to an encoder in order to tell whether a forward channel data was correctly decoded or not.

BT	URF	TR	ELNUM1	ELNUM	BCPM	BSB1	BEPB1	GN/MBA	BEPB2	RTR	BSTUF

FIGURE N.4/H.263

Structure of back-channel message (BCM) syntax for NEWPRED

N.4.2.1 Back-channel message Type (BT) (2 bits)

Back-channel message type indicates if the corresponding part of the encoded message is correctly decoded or not. Which type of message is required for the encoder is indicated in the picture header of the forward channel.

00: reserved for future use
01: reserved for future use
10: NACK. This indicates the erroneous decoding of the corresponding part of the forward channel data.
11: ACK. This indicates the correct decoding of the corresponding part of the forward channel data.

N.4.2.2 Unreliable Flag (URF) (1 bit)

The Unreliable Flag is set to 1 when a reliable value for TR or GN/MBA is not available to the decoder. (When BT is NACK, a reliable TR may not be available at the decoder.)

0: reliable
1: unreliable

N.4.2.3 Temporal Reference (TR) (10 bits)

Temporal reference contains the TR information of the video picture segment for which the ACK/NACK is indicated in the back-channel message. Note: The meaning of the term "video picture segment" as used herein is defined in Annex R. If a custom picture clock frequency was not in use for the reference picture, the two MSBs of TR are zero and the LSBs contain the eight bit TR found in the picture header of the reference picture. If a custom

picture clock frequency was in use for the reference picture, TR is a ten bit number consisting of the concatenation of ETR and TR from the reference picture header.

N.4.2.4 Enhancement Layer Number Indication (ELNUMI) (1 bit)

The enhancement layer number indication is "0" unless the optional Temporal, SNR and Spatial Scalability mode (Annex O) is used in the forward channel data and some enhancement layers of the forward channel are combined on one logical channel and the back-channel message refers to an enhancement layer (rather than the base layer), in which case the enhancement layer number indication shall be "1".

N.4.2.5 Enhancement Layer Number (ELNUM) (4 bits)

Enhancement layer number is present if and only if ELNUMI is "1", in which case it contains the layer number of the enhancement layer referred to in the back-channel message.

N.4.2.6 BCPM (1 bit)

BCPM is "0" unless the CPM mode is used in the forward channel data, in which case it is "1". If BCPM is "1", this indicates that BSBI is present.

N.4.2.7 Back-channel Sub Bitstream Indicator (BSBI) (2 bits)

A fixed length codeword of 2 bits that is only present if BCPM is "1". The BSBI is the natural binary representation of the appropriate Sub-Bitstream number in the forward channel data for which the ACK/NACK message is indicated in the back-channel message as described in section 5.2.4 and Annex C.

N.4.2.8 Back-channel Emulation Prevention Bit 1 (BEPB1) (1 bit)

A field which is present if and only if the videomux mode is in use. This field is always set to "1" to prevent a start code emulation.

N.4.2.9 GOB Number/Macroblock Address (GN/MBA) (5/6/7/9/11/12/13/14 bits):

A GOB number or Macroblock address is present in this field. If the optional Slice Structured mode (see Annex K) is not in use, this field contains the GOB number of the beginning of the video picture segment for which the NACK/ACK message is indicated in the back-channel message. If the optional Slice Structured mode is in use, this field contains the macroblock address of the beginning of the slice for which the NACK/ACK message is indicated in the back-channel message. The length of this field is the length specified elsewhere in this Recommendation for GN or MBA.

N.4.2.10 Back-channel Emulation Prevention Bit 2 (BEPB2) (1 bit)

A field which is present if and only if the videomux mode is in use. This field is always set to "1" to prevent a start code emulation.

N.4.2.11 Requested Temporal Reference (RTR) (10 bits)

Requested temporal reference is present only if BT is NACK. RTR indicates the requested temporal reference of the GOB or slice associated with the NACK. Typically it is the TR of the last correctly decoded video picture segment of the corresponding position at the decoder. If a custom picture clock frequency was not in use for the requested reference picture, the two MSBs of RTR are zero and the LSBs contain the eight bit TR found in the picture header of the requested reference picture. If a custom picture clock frequency was in use for the requested reference picture, RTR is a ten bit number consisting of the concatenation of ETR and TR from the requested reference picture header.

N.4.2.12 Stuffing (BSTUF) (Variable Length)

This field is present if and only if the separate logical channel mode is in use and the back channel message is the last in an external frame. BSTUF consists of a codeword of variable length consisting of zero or more bits of value "0". This field is only present at the end of an external frame.

N.5 Decoder Process

The decoder of this mode may need an additional number of picture memories to store the correctly decoded video signals and their Temporal Reference (TR) information. The decoder uses the stored picture for which the TR is TRP as the reference picture for inter-frame decoding instead of the last decoded picture, if the TRP field exists in the forward channel data. When the picture for which the TR is TRP is not available at the decoder, the decoder may send a forced INTRA update signal to the encoder by external means (for example Recommendation H.245).

Unless a different frame storage policy is negotiated by external means, correctly decoded video picture segments shall be stored into memory for use as later reference pictures on a first-in, first-out basis as shown in Figure N.1 (except for B pictures, which are not used as reference pictures), and video picture segments which are detected as having been incorrectly decoded should not replace correctly decoded ones in this memory area.

An acknowledgment message (ACK) and a non-acknowledgment message (NACK) are defined as back-channel messages. An ACK may be returned when the decoder decodes a video picture segment successfully. NACKs may be returned when the decoder fails to decode a video picture segment, and may continue to be returned until the decoder gets the expected forward channel data which includes the requested TRP or an INTRA update. Which types of message shall be sent is indicated in the RPSMF field of the picture header of the forward channel data.

In a usage scenario known as "Video Redundancy Coding", the Reference Picture Selection mode may be used by some encoders in a manner in which more than one representation is sent for the pictured scene at the same temporal instant (usually using different reference pictures). In such a case in which the Reference Picture Selection mode is in use and in which adjacent pictures in the bitstream have the same temporal reference, the decoder shall regard this occurrence as an indication that redundant copies have been sent of approximately the same pictured scene content, and shall decode and use the first such received picture while discarding the subsequent redundant picture(s).

Annex O
Temporal, SNR, and Spatial Scalability mode

(This annex forms an integral part of this Recommendation.)

This annex describes the optional mode of H.263 in support of Temporal, SNR, and Spatial Scalability. This mode may also be used in conjunction with error control schemes. The capability of this mode and the extent to which its features are supported is signaled by external means (for example Recommendation H.245). The use of this mode is indicated in PLUSPTYPE.

O.1 Overview

Scalability allows for the decoding of a sequence at more than one quality level. This is done by using a hierarchy of pictures and enhancement pictures partitioned into one or more layers. There are three types of pictures used for scalability: B, EI, and EP pictures, as explained below. Each of these has an enhancement layer number ELNUM that indicates to which layer it belongs, and a reference layer number RLNUM that indicates which layer is used for its prediction. The lowest layer is called the base layer, and has layer number 1.

Scalability is achieved by three basic methods: temporal, SNR, and spatial enhancement.

O.1.1 Temporal scalability

Temporal scalability is achieved using bi-directionally predicted pictures, or B-pictures. B-pictures allow prediction from either or both a previous and subsequent reconstructed picture in the reference layer. This property generally results in improved compression efficiency as compared to that of P pictures. These B-pictures differ from the B-picture part of a PB (or Improved PB) frame (see Annexes G and M) in that they are separate entities in the bitstream: they are not syntactically intermixed with a subsequent P (or EP) picture.

B pictures (and the B part of PB or Improved PB frames) are not used as reference pictures for the prediction of any other pictures. This property allows for B-pictures to be discarded if necessary without adversely affecting any subsequent pictures, thus providing temporal scalability. Figure O.1 illustrates the predictive structure of P and B pictures.

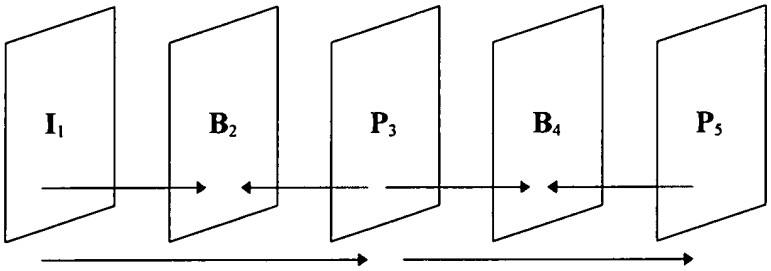


FIGURE O.1/H.263

Illustration of B picture prediction dependencies

The location of B pictures in the bitstream is in a data-dependence order rather than in strict temporal order. (This rule is consistent with the ordering of other pictures in the bitstream, but for all picture types other than the B picture, no such conflict arises between the data-dependence order and the temporal order.) For example, if the pictures of a video sequence were numbered 1, 2, 3, ..., then the bitstream order of the encoded pictures would be $I_1, P_3, B_2, P_5, B_4, \ldots$, where the subscript refers to the original picture number (as illustrated in Figure O.1).

There is no limit to the number of B pictures that may be inserted between pairs of reference pictures in the reference layer (other than what is necessary to prevent temporal ambiguity from overflows of the temporal reference field in the picture header). However, a maximum number of such pictures may be signaled by external means (for example Recommendation H.245).

The picture height, width, and pixel aspect ratio of a B picture shall always be equal to those of its temporally-subsequent reference layer picture.

Motion vectors are allowed to extend beyond the picture boundaries of B pictures.

O.1.2 SNR scalability

The other basic method to achieve scalability is through spatial/SNR enhancement. Spatial scalability and SNR scalability are equivalent except for the use of interpolation as is described shortly. Because compression introduces artifacts and distortions, the difference between a reconstructed picture and its original in the encoder is (nearly always) a nonzero-valued picture, containing what can be called the coding error. Normally, this coding error is lost at the encoder and never recovered. With SNR scalability, these coding error pictures can also be encoded and sent to the decoder, producing an enhancement to the decoded picture. The extra data serves to increase the signal-to-noise ratio of the video picture, and hence the term SNR scalability. Figure O.2 illustrates the data flow for SNR scalability. The vertical arrows from the lower layer illustrate that the picture in the enhancement layer is predicted from a reconstructed approximation of that picture in the reference (lower) layer.

If prediction is only formed from the lower layer, then the enhancement layer picture is referred to as an EI picture. It is possible, however, to create a modified bi-directionally predicted picture using both a prior enhancement layer picture and a temporally simultaneous lower layer reference picture. This type of picture is referred to as an EP picture or "Enhancement" P-picture. The prediction flow for EI and EP pictures is shown in Figure O.2. (Although not specifically shown in Figure O.2, an EI picture in an enhancement layer may have a P picture as its lower layer reference picture, and an EP picture may have an I picture as its lower-layer enhancement picture.)

For both EI and EP pictures, the prediction from the reference layer uses no motion vectors. However, as with normal P pictures, EP pictures use motion vectors when predicting from their temporally-prior reference picture in the same layer.

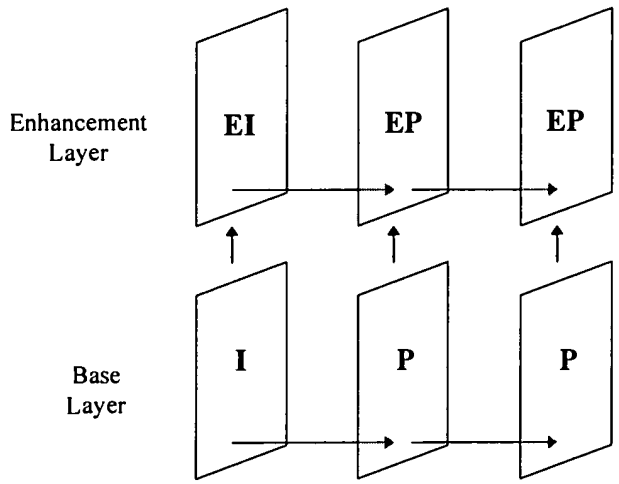


FIGURE O.2/H.263

Illustration of SNR scalability

O.1.3 Spatial scalability

The third and final scalability method in the Temporal, SNR, and Spatial Scalability mode is spatial scalability, which is closely related to SNR scalability. The only difference is that before the picture in the reference layer is used to predict the picture in the spatial enhancement layer, it is interpolated by a factor of two either horizontally or vertically (1-D spatial scalability), or both horizontally and vertically (2-D spatial scalability). The interpolation filters for this operation are defined in section O.6. For a decoder to be capable of some forms of spatial

scalability, it may also need to be capable of custom picture formats. For example, if the base layer is sub-QCIF (128x96), the 2-D spatial enhancement layer picture would be 256x192, which does not correspond to a standard picture format. Another example would be if the base layer were QCIF (176x144), with the standard pixel aspect ratio of 12:11. A 1-D horizontal spatial enhancement layer would then correspond to a picture format of 352x144 with a pixel aspect ratio of 24:11. Thus a custom picture format would have to be used for the enhancement layer in these cases. An example which does not require a custom picture format would be the use of a QCIF base layer with a CIF 2-D spatial enhancement layer. Spatial scalability is illustrated in Figure O.3.

Other than requiring an upsampling process to increase the size of the reference layer picture prior to its use as a reference for the encoding process, the processing and syntax for a spatial scalability picture is functionally identical to that for an SNR scalability picture.

Since there is very little syntactical distinction between pictures using SNR scalability and pictures using spatial scalability, the pictures used for either purpose are called EI and EP pictures.

The picture in the base layer which is used for upward prediction in an EI or EP picture may be an I picture, a P picture, or the P part of a PB or Improved PB frame (but shall not be a B picture or the B part of a PB or Improved PB frame).

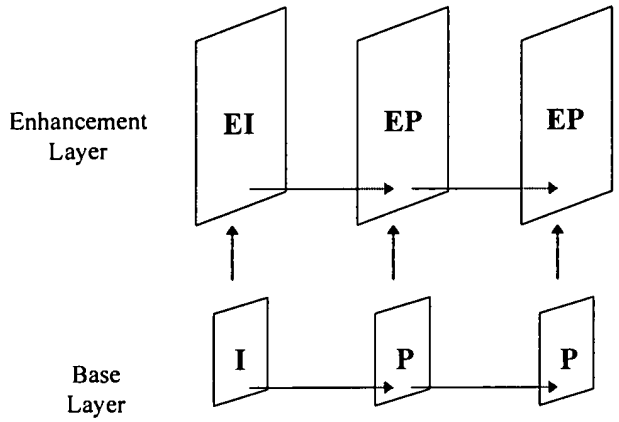


FIGURE O.3/H.263

Illustration of Spatial Scalability

O.1.4 Multilayer scalability

It is possible not only for B pictures to be temporally inserted between pictures of types I, P, PB, and Improved PB, but also between pictures of types EI and EP, whether these consist of SNR or spatial enhancement pictures. It is also possible to have more than one SNR or spatial enhancement layer in conjunction with a base layer. Thus a multilayer scalable bitstream can be a combination of SNR layers, spatial layers, and B pictures. The size of a picture cannot decrease, however, with increasing layer number. It can only stay the same or increase by a factor of two in one or both dimensions. Figure O.4 illustrates a multilayer scalable bitstream.

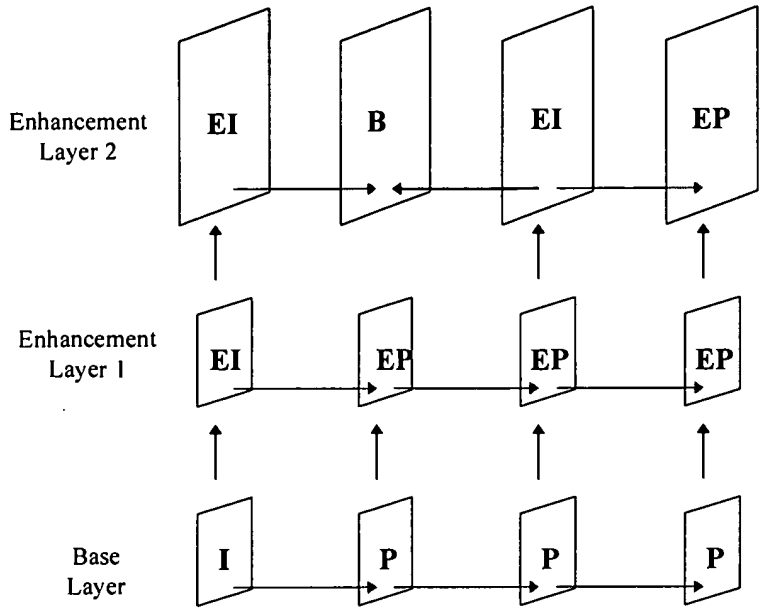


FIGURE O.4/H.263

Illustration of multilayer scalability

In the case of multilayer scalability, the picture in a reference layer which is used for upward prediction in an EI or EP picture may be a I, P, EI, or EP picture, or may be the P part of a PB or Improved PB frame in the base layer (but shall not be a B picture or the B part of a PB or Improved PB frame).

As with the two layer case, B pictures may occur in any layer. However, any picture in an enhancement layer which is temporally simultaneous with a B picture in its reference layer must be a B picture or the B picture part of an PB or Improved PB frame. This is to preserve the disposable nature of B pictures. Note, however, that B pictures may occur in layers that have no corresponding picture in lower layers. This allows an encoder to send enhancement video with a higher picture rate than the lower layers.

The enhancement layer number and the reference layer number for each enhancement picture (B, EI, or EP) are indicated in the ELNUM and RLNUM fields, respectively, of the picture header (when present). See the inference rules described in section 5.1.4.4 for when these fields are not present. If a B picture appears in an enhancement layer in which temporally-surrounding SNR or Spatial Scalability pictures also appear, the reference layer number (RLNUM) of the B picture shall be the same as the enhancement layer number (ELNUM).

The picture height, width, and pixel aspect ratio of a B picture shall always be equal to those of its temporally-subsequent reference layer picture.

O.2 Transmission order of pictures

Pictures which are dependent on other pictures shall be located in the bitstream after the pictures on which they depend.

The bitstream syntax order is specified such that for reference pictures (i.e., picture having types I, P, EI, or EP, or the P part of PB or Improved PB), the following two rules shall be obeyed:

1. All reference pictures with the same temporal reference shall appear in the bitstream in increasing enhancement layer order (since each lower layer reference picture is needed to decode the next higher layer reference picture).

2. All temporally simultaneous reference pictures as discussed in item 1 above shall appear in the bitstream prior to any B pictures for which any of these reference pictures is the first temporally subsequent reference picture in the reference layer of the B picture (in order to reduce the delay of decoding all reference pictures which may be needed as references for B pictures).

Then, the B-pictures with earlier temporal references shall follow (temporally ordered within each enhancement layer).

The bitstream location of each B picture shall comply with the following rules:

1. Its bitstream location shall be after that of its first temporally subsequent reference picture in the reference layer (since the decoding of the B picture generally depends on the prior decoding of that reference picture).
2. Its bitstream location shall be after that of all reference pictures that are temporally simultaneous with the first temporally subsequent reference picture in the reference layer (in order to reduce the delay of decoding all reference pictures which may be needed as references for B pictures)
3. Its bitstream location shall precede the location of any additional temporally subsequent pictures other than B pictures in its reference layer (since to allow otherwise would increase picture-storage memory requirements for the reference layer pictures).
4. Its bitstream location shall be after that of all EI and EP pictures that are temporally simultaneous with the first temporally subsequent reference picture.
5. Its bitstream location shall precede the location of all temporally subsequent pictures within its same enhancement layer (since to allow otherwise would introduce needless delay and increase picture-storage memory requirements for the enhancement layer).

Figure O.5 illustrates two allowable picture transmission orders given by the rules above for the layering structure shown therein (with numbers in dotted-line boxes indicating the bitstream order, separated by commas for the two alternatives).

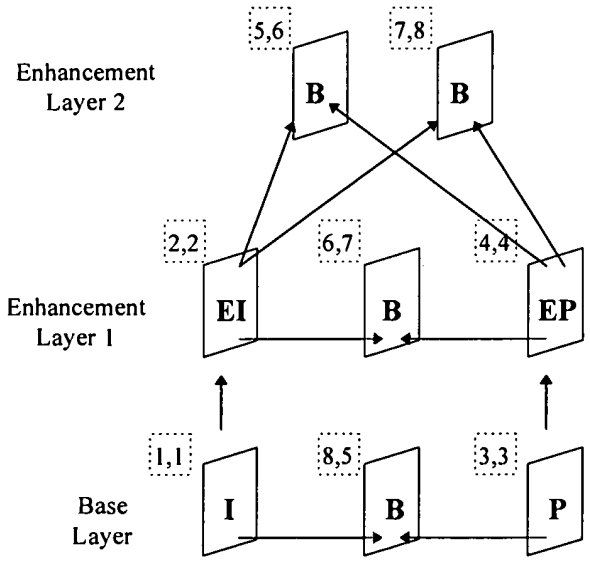


FIGURE O.5/H.263

Example of picture transmission order

O.3 Picture Layer Syntax

The Enhancement Layer Number (ELNUM, section 5.1.11) is always present in any enhancement (B, EI, or EP) picture and shall not be present in I or P pictures, or in PB or Improved PB frames. The Reference Layer Number (RLNUM, section 5.1.12) is present for some enhancement pictures and is inferred for others, as described in section 5.1.12.

There is exactly one base layer and it has ELNUM and RLNUM equal to 1. The RLNUM gives the enhancement layer number of the forward and backward reference pictures for B-pictures, for the upward reference picture for EI and EP-pictures. The reference pictures of the base layer may consist of I, PB, Improved PB, and P pictures, none of which include ELNUM or RLNUM in the picture header (their implied values are 1).

For B-pictures, RLNUM must be less or equal to ELNUM, while for EI and EP-pictures, RLNUM must be smaller than ELNUM.

ELNUM may be different from the layer number used at the system level. Since B-pictures have no other pictures dependent on themselves, they may even be put in a separate enhancement layer by system components external to H.263 (for example Recommendations H.245 and H.223). Moreover, it is up to the implementor as to whether the enhancement pictures are sent in separate video channels or remain multiplexed together with the base layer pictures.

As stated in section 5.1.4.5, the Deblocking Filter mode (see Annex J) does not apply within B pictures. This is because B pictures are not used for the prediction of any other pictures, and thus the application of a deblocking filter to these pictures is entirely a post-processing technique which is outside the scope of this Recommendation. However, the use of some type of deblocking filter for B pictures is encouraged, and a filter such as that described in Annex J may in fact be a good design to use for that purpose.

The Temporal Reference (TR) (see section 5.1.2) is defined exactly as for I and P pictures.

There shall be no TR_B (see section 5.1.22) or DBQUANT (see section 5.1.23) fields in the picture header of B, EI or EP-pictures.

O.4 Macroblock Layer Syntax

The macroblock layer syntax for B and EP-pictures is the same, since each uses two reference pictures in a similar way. However, the interpretation varies slightly depending on the picture type. Figure O.6 indicates the B and EP syntax. The MBTYPE field indicates whether there is direct-mode prediction, forward prediction, backward/upward prediction, or bi-directional prediction. The MBTYPE is defined differently for B and EP-pictures as described below.

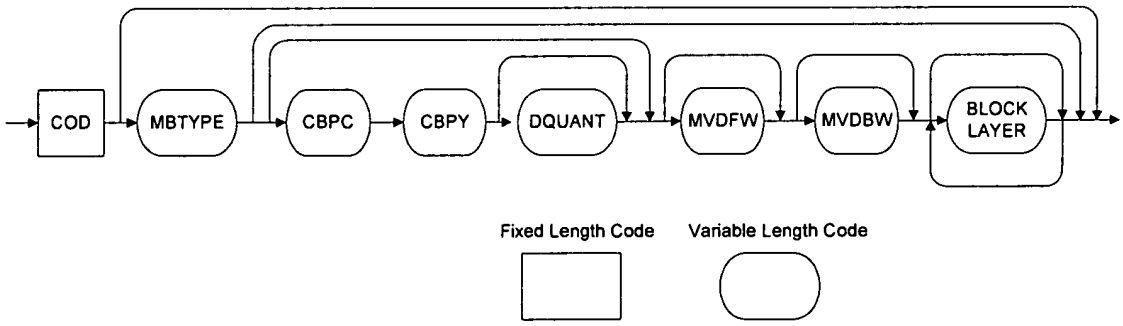


FIGURE O.6/H.263

Macroblock syntax for EP and B pictures

The direct prediction mode is only available for B-pictures. It is a bi-directional prediction mode similar to the bi-directional mode in Improved PB-frames mode (Annex M). The only difference is that there is no restriction on which pixels can be backward predicted since the complete backward prediction image is known at the decoder. Bi-directional mode uses separate motion vectors for forward and backward prediction. In both direct and bi-directional mode, the prediction pixel values are calculated by averaging the forward and backward prediction pixels. The average is calculated by dividing the sum of the two predictions by two (division by truncation). In direct mode, when there are four motion vectors in the reference macroblock, then all four motion vectors are used as in Improved PB-frames mode (Annex M).

For B pictures, forward prediction means prediction from a previous reference picture in the reference layer. Backward prediction means prediction from a temporally subsequent reference picture in the reference layer.

For EP-pictures, forward prediction means prediction from a previous EI or EP picture in the same layer, while upward prediction is used to mean prediction from the temporally simultaneous (possibly interpolated) reference picture in the reference layer. No motion vector is used for the upward prediction (which is syntactically in the same place as backward prediction for B pictures), although one can be used for the forward prediction.

The macroblock syntax for EI-pictures is slightly different. As shown in Figure O.7, the MBTYPE and CBPC are combined into an MCBPC field. No forward prediction is used, only upward prediction from the temporally simultaneous reference picture in the reference layer. No motion vector is used.

In B and EP pictures, motion vectors over picture boundaries may be used as described in section D.1 of Annex D (although extension of the motion vector range as described in section D.2 is only active if the Unrestricted Motion Vector mode is also in use).

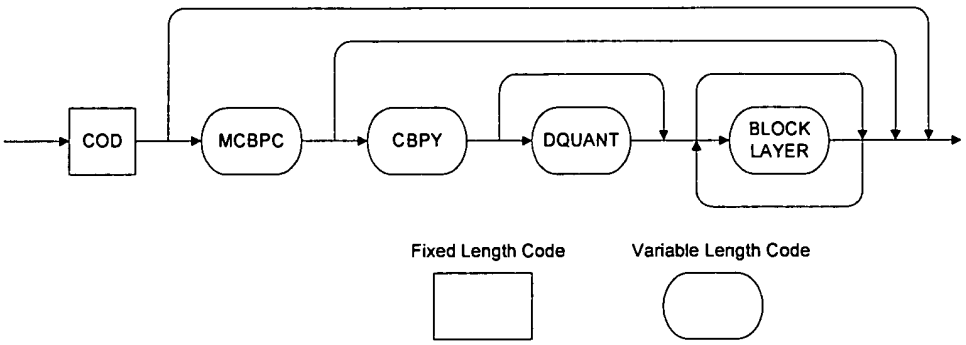


FIGURE O.7/H.263

Macroblock syntax for EI pictures

O.4.1 Coded macroblock indication (COD) (1 bit)

A bit which when set to "0" signals that the macroblock is coded. If set to "1", no further information is transmitted for this macroblock and the macroblock is handled as "skipped" as described below.

O.4.2 MBTYPE/MCBPC (VLC)

There are different MBTYPE tables for B and EP pictures. For EI pictures, there is instead a MCBPC table. Table O.1 is the MBTYPE table for B-pictures. Table O.2 is the MBTYPE table for EP pictures. Table O.3 is the MCBPC table for EI pictures.

For B pictures, the "Direct (skipped)" prediction type indicates that neither MBTYPE, nor any data is transmitted in the macroblock and that the decoder derives forward and backward motion vectors and the corresponding bi-directional prediction. This is signaled by the COD bit. The "Forward (no texture)", the "Backward (no texture)", and the "Bi-dir (no texture)" prediction types for a B picture indicate forward, backward, and bi-directional prediction with no coefficients, and with one transmitted motion vector for forward and backward prediction, and two transmitted motion vectors for bi-directional prediction.

For EP pictures, the "Forward (skipped)" prediction type indicates that no additional data is sent for the macroblock, so that the decoder should use forward prediction with a zero motion vector and no coefficients. The "Upward (no texture)" and "Bi-dir (no texture)" prediction types for an EP picture indicate upward and bi-directional prediction with no coefficients, and with zero motion vector(s).

For EI pictures, the "Upward (skipped)" prediction type indicates that no additional data is sent for the macroblock, so that the decoder should use upward prediction with a zero motion vector and no coefficients.

TABLE O.1/H.263

MBTYPE VLC codes for B pictures

Index	Prediction Type	MVDFW	MVDBW	CBPC + CBPY	DQUANT	MBTYPE	Bits
—	Direct (skipped)					(COD=1)	0
0	Direct			X		11	2
1	Direct + Q			X	X	0001	4
2	Forward (no texture)	X				100	3
3	Forward	X		X		101	3
4	Forward + Q	X		X	X	0011 0	5
5	Backward (no texture)		X			010	3
6	Backward		X	X		011	3
7	Backward + Q		X	X	X	0011 1	5
8	Bi-Dir (no texture)	X	X			0010 0	5
9	Bi-Dir	X	X	X		0010 1	5
10	Bi-Dir + Q	X	X	X	X	0000 1	5
11	INTRA			X		0000 01	6
12	INTRA + Q			X	X	0000 001	7
13	Stuffing					0000 0000 1	9

TABLE O.2/H.263

MBTYPE VLC codes for EP pictures

Index	Prediction Type	MVDFW	MVDBW	CBPC + CBPY	DQUANT	MBTYPE	Bits
—	Forward (skipped)					(COD=1)	0
0	Forward	X		X		1	1
1	Forward + Q	X		X	X	001	3
2	Upward (no texture)					010	3
3	Upward			X		011	3
4	Upward + Q			X	X	0000 1	5
5	Bi-Dir (no texture)					0001 0	5
6	Bi-Dir	X		X		0001 1	5
7	Bi-Dir + Q	X		X	X	0000 01	6
8	INTRA			X		0000 001	7
9	INTRA + Q			X	X	0000 0001	8
10	Stuffing					0000 0000 1	9

TABLE O.3/H.263

MCBPC VLC codes for EI pictures

Index	Prediction Type	Code block pattern (56)	CBPY	DQUANT	MCBPC	Bits
—	Upward (skipped)				(COD = 1)	0
0	Upward	00	X		1	1
1	Upward	01	X		001	3
2	Upward	10	X		010	3
3	Upward	11	X		011	3
4	Upward + Q	00	X	X	0001	4
5	Upward + Q	01	X	X	0000 001	7
6	Upward + Q	10	X	X	0000 010	7
7	Upward + Q	11	X	X	0000 011	7
8	INTRA	00	X		0000 0001	8
9	INTRA	01	X		0000 1001	8
10	INTRA	10	X		0000 1010	8
11	INTRA	11	X		0000 1011	8
12	INTRA+Q	00	X	X	0000 1100	8
13	INTRA+Q	01	X	X	0000 1101	8
14	INTRA+Q	10	X	X	0000 1110	8
15	INTRA+Q	11	X	X	0000 1111	8
16	Stuffing				0000 0000 1	9

O.4.3 Coded Block Pattern for Chrominance (CBPC) (Variable Length)

When present, CBPC indicates the coded block pattern for chrominance blocks as described in Table O.4. CBPC is present only for EP and B pictures, when its presence is indicated by MBTYPE (see Tables O.1 and O.2).

TABLE O.4/H.263

CBPC VLC codes

Index	Code block pattern (56)	CBPC	Bits
0	00	0	1
1	01	10	2
2	10	111	3
3	11	110	3

O.4.4 Coded Block Pattern for Luminance(CBPY) (Variable Length)

When present, CBPY indicates which blocks in the luminance portion of the macro block are present. CBPY is present only when its presence is indicated by MBTYPE (see Tables O.1, O.2 and O.3). CBPY is coded as described in section 5.3.5 and Table 13. Upward predicted macroblocks in EI and EP pictures, bi-directional predicted macroblocks in EP pictures, and INTRA macroblocks in EI, EP, and B pictures use the CBPY definition for INTRA macroblocks, and other macroblock types in EI, EP, and B pictures use the CBPY definition for INTER macroblocks.

O.4.5 Quantizer Information (DQUANT) (2 bits/Variable Length)

DQUANT is used as in other picture macroblock types. See section 5.3.6 and Annex T.

O.4.6 Motion vector data (MVDFW, MVDBW) (Variable Length)

MVDFW is the motion vector data for the forward vector, if present. MVDBW is the motion vector data for the backward vector, if present (allowed only in B pictures). The variable length codewords are given in Table 14, or Table D.3 if the Unrestricted Motion Vector mode is used (see Annex D).

O.5 Motion vector decoding

O.5.1 Differential Motion Vectors

Motion vectors for forward, backward or bi-directionally predicted blocks are differentially encoded. To recover the macroblock motion vectors, a prediction is added to the motion vector differences. The predictions are formed in a similar manner to that described in section 6.1.1, except that forward motion vectors are predicted only from forward motion vectors in surrounding macroblocks, and backward motion vectors are predicted only from backward motion vectors in surrounding macroblocks. The same decision rules apply for the special cases at picture, GOB or slice borders as described in section 6.1.1. If a neighboring macroblock does not have a motion vector of the same type (forward or backward), the candidate predictor for that macroblock is zero for that motion vector type.

O.5.2 Motion vectors in direct mode

For macroblocks coded in direct mode, no vector differences are transmitted. Instead, the forward and backward motion vectors are directly computed from the temporally consecutive P-vector as described in G.4 with the restriction that MV_D is always zero. These derived vectors are not used for prediction of other motion vectors.

O.6 Interpolation filters

The method by which a picture is interpolated for 2-D spatial scalability is shown in Figures O.8 and O.9. The first figure shows the interpolation for interior pixels, while the second one shows the interpolation close to picture boundaries. This is the same technique used in Annex Q and in some cases in Annex P. The method by which a picture is interpolated for 1-D spatial scalability is shown in Figures O.10 and O.11. Figure O.10 shows the interpolation for interior pixels in the horizontal direction. The interpolation for the vertical direction is analogous. Figure O.11 shows the interpolation for pixels at picture boundaries. Again, the interpolation for the vertical direction is analogous. Again, this is the same technique use in some cases in Annex P.

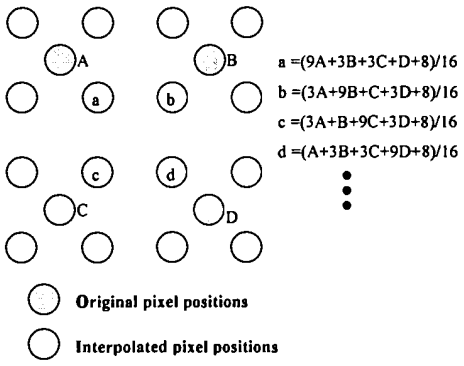


FIGURE O.8/H.263

Method for interpolating pixels for 2-D scalability

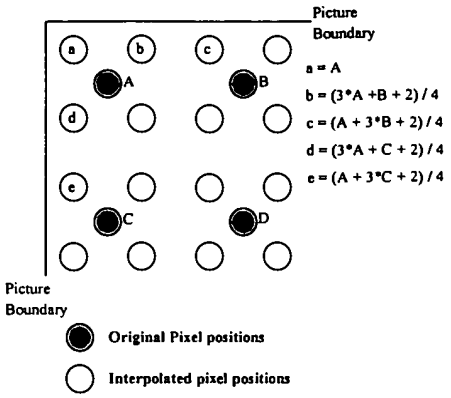


FIGURE O.9/H.263

Method for 2-D interpolation at boundaries

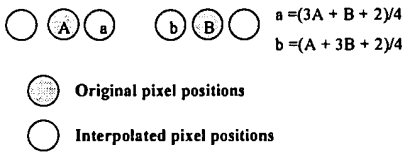


FIGURE O.10/H.263

Method for interpolating pixels for 1-D scalability

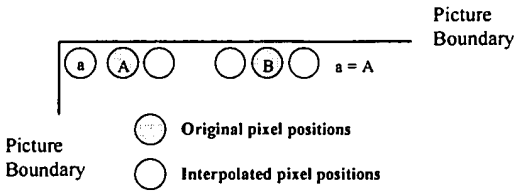


FIGURE O.11/H.263

Method for 1-D interpolation at picture boundaries.

Annex P
Reference Picture Resampling

(This annex forms an integral part of this Recommendation.)

P.1 Introduction

This annex describes the use and syntax of a resampling process which can be applied to the previous decoded reference picture in order to generate a "warped" picture for use in predicting the current picture. This resampling syntax can specify the relationship of the current picture to a prior picture having a different source format, and can also specify a "global motion" warping alteration of the shape, size, and location of the prior picture with respect to the current picture. In particular, the Reference Picture Resampling mode can be used to adaptively alter the resolution of pictures during encoding. A fast algorithm is used to generate bilinear interpolation coefficients. The capability to use this mode and the extent to which its features are supported is negotiated externally (for example Recommendation H.245). This mode may be used in restricted scenarios that may be defined during capability negotiation (e.g., to support only factor-of-four picture resizing, to support only half-pixel resolution picture warping, or to support arbitrary image size resizing and displacements).

Note: The default image transformation between pictures of differing resolutions is defined to maintain the spatial alignment of the edges of the picture area and of the relative locations of the luminance and chrominance samples. This may have implications on the design of any resampling operations used for generating pictures of various resolutions at the encoder and for displaying pictures of various resolutions after decoding (particularly regarding shifts in spatial location caused by phase shifts induced in resampling). Also, since this mode can be used for dynamic adaptive picture resolution changes, operation with this mode may benefit from external negotiation to display the decoded picture at a higher resolution than its encoded picture size, in order to allow switching between encoded picture sizes without resizing the picture display.

If the Reference Picture Resampling bit in the PLUSPTYPE field is not set, but PLUSPTYPE is present and the picture is an INTER, B, or EP picture or an Improved PB frame, and the picture size differs from that of the temporally previous encoded picture, this condition invokes Reference Picture Resampling with warping parameters (see section P.2.2) set equal to zero, fill mode (see section P.2.3) set to *clip*, and displacement accuracy (see section P.2.1) set to $\frac{1}{16}$-pixel accuracy. This causes the resampling process to easily act as a predictively-encoded change in picture resolution. In simple factor-of-four resolution change cases such as transitions between CIF and 4CIF, the resampling process reduces to the same simple filter used for Spatial Scalability (Annex O) or for Reduced Resolution Update (Annex Q), except for the application of rounding control.

If the picture is an EP picture and the Reference Picture Resampling bit is set in the PLUSPTYPE field of the picture header of the reference layer, this bit shall also be set in the picture header of the EP picture in the enhancement layer.

If a B-picture uses the Reference Picture Resampling mode, the resampling process shall be applied to the temporally previous anchor picture and not to the temporally subsequent one. The temporally previous anchor picture to which the resampling process is applied shall be the decoded picture (i.e., prior to any resampling applied by Reference Picture Resampling if this mode is also invoked for the subsequent reference picture). The temporally subsequent anchor picture shall have the same picture size as the B picture.

If the Reference Picture Resampling mode is invoked for an Improved PB frame, one set of warping parameters is sent and the resampled reference picture is used as the reference for both the B and P part of the Improved PB frame.

The Reference Picture Resampling is defined in terms of the displacement of the four corners of the current picture area. For the luminance field of the current picture with horizontal size H and vertical size V, four conceptual motion vectors $\underline{v}^{00}$, $\underline{v}^{H0}$, $\underline{v}^{0V}$, and $\underline{v}^{HV}$ are defined for the upper left, upper right, lower left, and lower right corners of the picture, respectively. These vectors describe how to move the corners of the current picture to map them onto the corresponding corners of the previous decoded picture, as shown in Figure P.1. The units of these vectors are the same as those in the reference picture grid. To generate a vector $\underline{v}(x,y)$ at some real-valued location (x, y) in the interior of the current picture, an approximation to bilinear interpolation is used, *i.e.*, as in

$$\underline{v}(x,y) = \left(1-\frac{y}{V}\right)\left[\left(1-\frac{x}{H}\right)\underline{v}^{00} + \left(\frac{x}{H}\right)\underline{v}^{H0}\right] + \left(\frac{y}{V}\right)\left[\left(1-\frac{x}{H}\right)\underline{v}^{0V} + \left(\frac{x}{H}\right)\underline{v}^{HV}\right].$$

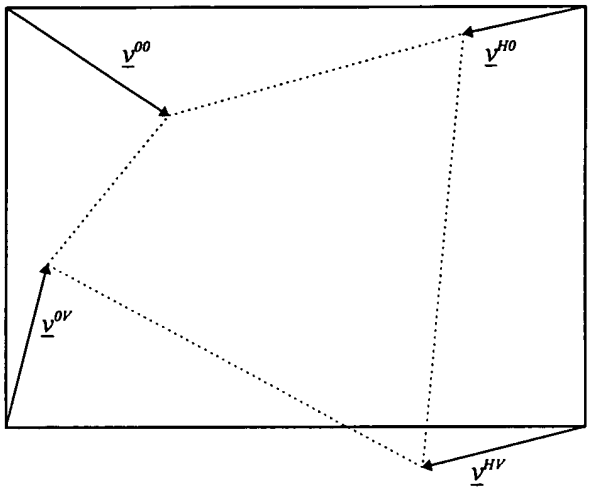


FIGURE P.1/H.263

Example of conceptual motion vectors used for warping

The horizontal size H and vertical size V of the current picture, and the horizontal size H_R and vertical size V_R of the reference picture are those indicated by the picture header, regardless of whether these values are divisible by 16 or not. If the picture width or height is not divisible by 16, the additional area shall be generated by adding pixels to the resampled picture using the same fill mode as used in the resampling process.

For the simplicity of description, the resampling vectors $\underline{r}^0$, $\underline{r}^x$, $\underline{r}^y$, and $\underline{r}^{xy}$ are defined as

$$\underline{r}^0 = \underline{v}^{00}$$

$$\underline{r}^x = \underline{v}^{H0} - \underline{v}^{00}$$

$$\underline{r}^y = \underline{v}^{0V} - \underline{v}^{00}$$

$$\underline{r}^{xy} = \underline{v}^{00} - \underline{v}^{H0} - \underline{v}^{0V} + \underline{v}^{HV} .$$

Using this definition, the equation for bilinear interpolation is rewritten as

$$\underline{v}(x,y) = \underline{r}^0 + \left(\frac{x}{H}\right)\underline{r}^x + \left(\frac{y}{V}\right)\underline{r}^y + \left(\frac{x}{H}\right)\left(\frac{y}{V}\right)\underline{r}^{xy}$$

For warping purposes, it is assumed that the coordinates of the upper left corner of the picture area are $(x, y)=(0,0)$ and that each pixel has unit height and width, so that the centers of the pixels lie at the points $(x, y)=(i_L + \frac{1}{2}, j_L + \frac{1}{2})$ for $i_L = 0, \ldots, H-1$ and $j_L = 0, \ldots, V-1$, where the L subscript indicates that i_L and j_L pertain to the luminance field. (As the pixel aspect ratio is usually constant or the conversion of the aspect ratio is performed by this resampling, there is no need to consider the actual pixel aspect ratio for these purposes.) Using this convention, the x- and y-displacements at the locations of interest in the luminance field of the reference picture are

$$v_x(i_L, j_L) = \frac{1}{HV}\left[HVr_x^0 + \left(i_L + \tfrac{1}{2}\right)Vr_x^x + \left(j_L + \tfrac{1}{2}\right)Hr_x^y + \left(i_L + \tfrac{1}{2}\right)\left(j_L + \tfrac{1}{2}\right)r_x^{xy}\right]$$

$$v_y(i_L, j_L) = \frac{1}{HV}\left[HVr_y^0 + \left(i_L + \tfrac{1}{2}\right)Vr_y^x + \left(j_L + \tfrac{1}{2}\right)Hr_y^y + \left(i_L + \tfrac{1}{2}\right)\left(j_L + \tfrac{1}{2}\right)r_y^{xy}\right]$$

Because all positions and phases must be computed relative to the center of the upper-left corner pixel, which has the coordinate $(x, y)=(\frac{1}{2},\frac{1}{2})$, the quantities of primary interest are

$$x_R(i_L,j_L)-\frac{1}{2}=\left(i_L+\frac{1}{2}\right)+v_x(i_L,j_L)-\frac{1}{2}$$

$$=\frac{1}{HV}\left[HVr_x^0+\left(i_L+\frac{1}{2}\right)(HV+Vr_x^x)+\left(j_L+\frac{1}{2}\right)Hr_x^y+\left(i_L+\frac{1}{2}\right)\left(j_L+\frac{1}{2}\right)r_x^{xy}\right]-\frac{1}{2}$$

$$y_R(i_L,j_L)-\frac{1}{2}=\left(j_L+\frac{1}{2}\right)+v_y(i_L,j_L)-\frac{1}{2}$$

$$=\frac{1}{HV}\left[HVr_y^0+\left(j_L+\frac{1}{2}\right)Vr_y^x+\left(j_L+\frac{1}{2}\right)(HV+Hr_y^y)+\left(i_L+\frac{1}{2}\right)\left(j_L+\frac{1}{2}\right)r_y^{xy}\right]-\frac{1}{2}$$

Once a location has been determined in the prior decoded reference picture using an approximation of these equations, bilinear interpolation as specified later in this annex shall be used to generate a value for the resampled pixel.

Each resampling vector can be decomposed into two components, with the first component describing the geometrical warping and the second component accounting for any difference in size between the predicted picture (horizontal size H and vertical size V) and the reference picture (horizontal size H_R and vertical size V_R). This decomposition is as follows:

$$\begin{aligned}
\underline{v}^{00} &= \underline{v}^{00}_{warp}+\underline{v}^{00}_{size} = \underline{v}^{00}_{warp}+(0,0)\\
\underline{v}^{H0} &= \underline{v}^{H0}_{warp}+\underline{v}^{H0}_{size} = \underline{v}^{H0}_{warp}+(H_R-H,0)\\
\underline{v}^{0V} &= \underline{v}^{0V}_{warp}+\underline{v}^{0V}_{size} = \underline{v}^{0V}_{warp}+(0,V_R-V)\\
\underline{v}^{HV} &= \underline{v}^{HV}_{warp}+\underline{v}^{HV}_{size} = \underline{v}^{HV}_{warp}+(H_R-H,V_R-V)
\end{aligned}$$

P.2 Syntax

Whenever the reference picture resampling bit is set in the PLUSTPYPE field of the picture header, the RPRP field of the picture header includes parameters which control the Reference Picture Resampling process. This includes a two-bit warping displacement accuracy (WDA) field, may include eight warping parameters or one-bit warping parameter refinements, and includes a fill mode, as described in this section.

P.2.1 Warping Displacement Accuracy (WDA) (2 bits)

A two bit warping displacement accuracy field WDA appears first in the RPRP field of the bitstream, and indicates the accuracy of the displacements for each pixel. A value of "10" indicates that the x- and y-displacements for each pixel are quantized to half-pixel accuracy. A value of "11" indicates that the displacements are quantized to $\frac{1}{16}$-pixel accuracy. The use of other values is reserved.

P.2.2 Warping Parameters (Variable Length)

When the Reference Picture Resampling parameters are sent for an INTER or B picture, or an Improved PB frame, eight warping parameters are included in the picture header using the variable length code (VLC) shown in Table D.3. For an EP picture using SNR scalability, the warping parameters of the lower layer are used and no warping parameters are transmitted. If the Reference Picture Resampling bit is set in the PLUSPTYPE field of the picture header for an EP picture using spatial scalability, the warping parameters of the lower layer are refined to the accuracy needed for the current layer by multiplying the warping parameter for each up-sampled dimension (warping parameters with subscript x and/or y) of the lower layer by two and adding the value of one additional bit that is sent in place of the associated warping parameter to define the least significant bit of the warping parameter.

The eight integer warping parameters (or their one-bit refinements) are sent in the following order:

$$w_x^0, w_y^0, w_x^x, w_y^x, w_x^y, w_y^y, w_x^{xy}, \text{ and } w_y^{xy}.$$

When not a one-bit refinement, these parameters are sent in a similar manner as motion vector differences in the Unrestricted Motion Vector mode when PLUSPTYPE is present, using Table D.3 with no restriction on the range of the parameters (i.e., a range of −4095 to +4095). As when encoding motion vector difference pairs, an emulation prevention bit shall be added as needed after each pair of warping parameters is sent, such that if the all-zero codeword (the value +1 in half-pixel units) of Table D.3 is used for both warping parameters in the pair (w_x^0 and w_y^0, w_x^x and w_y^y, or w_x^{xy} and w_y^{xy}), the pair of codewords is followed by a single bit equal to 1 to prevent start code emulation.

These eight warping parameters are interpreted as picture corner displacements relative to the displacements that would be induced by removing the resizing component of the resampling vectors. The warping parameters are scaled to represent half-pixel offsets in the luminance field of the current picture, and the range for the values of these paramenters is −4095 to +4095. The warping parameters are defined using the resampling vectors by the relations

$$w_x^0 = 2r_x^0 \qquad w_y^0 = 2r_y^0$$

$$w_x^x = 2\left(r_x^x - (H_R - H)\right) \qquad w_y^x = 2r_y^x$$

$$w_x^y = 2r_x^y \qquad w_y^y = 2\left(r_y^y - (V_R - V)\right)$$

$$w_x^{xy} = 2r_x^{xy} \qquad w_y^{xy} = 2r_y^{xy}$$

P.2.3 Fill Mode (FILL_MODE) (2 bits)

For an INTER or B picture or an Improved PB frame, immediately following the VLC-coded warping parameters in the picture header are two bits which define the fill-mode action to be taken for the values of pixels for which the calculated location in the reference picture lies outside of the reference picture area. The meaning of these two bits is shown in Table P.1 and their location is shown in Figure P.2. For an EP picture, the fill-mode action is the same as that for the reference layer, and the two fill-mode bits are not sent.

TABLE P.1/H.263

Fill mode bits/action

fill-mode bits	fill action
0 0	*color*
0 1	*black*
1 0	*gray*
1 1	*clip*

If the fill mode is *clip*, the coordinates of locations in the prior reference picture are independently limited as in the Unrestricted Motion Vector mode so that pixel values outside of the prior reference picture area are estimated by extrapolation from the values of pixels at the image border. If the fill mode is *black*, luminance samples outside of the prior reference picture area are assigned a value of $Y = 16$ and chrominance samples are assigned a value of $C_B = C_R = 128$. If the fill mode is *gray*, luminance and chrominance values are assigned a value of $Y = C_B = C_R = 128$. If the fill mode is *color*, then additional fields are sent to specify a fill color, as described in the next section.

P.2.4 Fill Color Specification (Y_FILL, C_B_EPB, C_B_FILL, C_R_EPB, C_R_FILL) (26 bits)

If the fill mode is *color* and the picture is not an EP picture, then the fill-mode bits are followed in the bitstream by three eight-bit integers, *Y_fill*, C_B*_fill*, and C_R*_fill*, which specify a fill color precisely. Between these three eight-

bit integers are two emulation prevention bits (C_B_EPB and C_R_EPB) each of which is equal to 1. The format of this color specification, which is present only when the fill mode is *color*, is shown in Figure P.2. Each eight-bit integer field is sent using its natural representation. For an EP picture, the fill-mode action (and fill color) is the same as that for the reference layer, and the fill color specification is not sent.

FIGURE P.2/H.263

Format of fill mode and fill color specification data

P.3 Resampling algorithm

The method described in this section shall be mathematically equal in result to that used to generate the samples of the resampled reference picture. Using the integer warping parameters $w_x^0, w_y^0, w_x^x, w_y^x, w_x^y, w_y^y, w_x^{xy}$, and w_y^{xy}, integer parameters $u_x^{00}, u_y^{00}, u_x^{H0}, u_y^{H0}, u_x^{0V}, u_y^{0V}, u_x^{HV}$, and u_y^{HV} which denote the x- and y-displacements at the corners of the luminance field in $\frac{1}{32}$-pixel accuracy (the actual displacements are obtained by dividing these values by 32) are defined as

$$u_x^{00} = 16w_x^0 \qquad u_y^{00} = 16w_y^0$$
$$u_x^{H0} = 16\left(w_x^0 + w_x^x + 2(H_R - H)\right) \qquad u_y^{H0} = 16\left(w_y^0 + w_y^x\right)$$
$$u_x^{0V} = 16\left(w_x^0 + w_x^y\right) \qquad u_y^{0V} = 16\left(w_y^0 + w_y^y + 2(V_R - V)\right)$$
$$u_x^{HV} = 16\left(w_x^0 + w_x^x + w_x^y + w_x^{xy} + 2(H_R - H)\right) \qquad u_y^{HV} = 16\left(w_y^0 + w_y^x + w_y^y + w_y^{xy} + 2(V_R - V)\right)$$

Next, H' and V', which denote the horizontal and vertical size of the *virtual frame*, are defined as the smallest integers that satisfy the following condition

$$H' \geq H,\ V' \geq V,\ H' = 2^m,\ V' = 2^n,\ m \text{ and } n \text{ are positive integers.}$$

By applying bilinear extrapolation to the corner vectors of the luminance field, the integer parameters $u_x^{LT}, u_y^{LT}, u_x^{RT}, u_y^{RT}, u_x^{LB}, u_y^{LB}, u_x^{RB}$, and u_y^{RB} which denote the x- and y-displacements of the luminance field at the *virtual points* $(x, y) = (0, 0)$, $(H', 0)$, $(0, V')$, and (H', V') in $\frac{1}{32}$-pixel accuracy (the actual displacements are obtained by dividing these values by 32) are defined as

$$u_x^{LT} = u_x^{00} \qquad u_y^{LT} = u_y^{00}$$
$$u_x^{RT} = \left((H - H')u_x^{00} + H'u_x^{H0}\right) // H \qquad u_y^{RT} = \left((H - H')u_y^{00} + H'u_y^{H0}\right) // H$$
$$u_x^{LB} = \left((V - V')u_x^{00} + V'u_x^{0V}\right) // V \qquad u_y^{LB} = \left((V - V')u_y^{00} + V'u_y^{0V}\right) // V$$
$$u_x^{RB} = \left((V - V')\left((H - H')u_x^{00} + H'u_x^{H0}\right) + V'\left((H - H')u_x^{0V} + H'u_x^{HV}\right)\right) // (HV)$$
$$u_y^{RB} = \left((V - V')\left((H - H')u_y^{00} + H'u_y^{H0}\right) + V'\left((H - H')u_y^{0V} + H'u_y^{HV}\right)\right) // (HV)$$

where "//" denotes integer division that rounds the quotient to the nearest integer, and rounds half integer values away from 0.

In the remainder of this annex, it is assumed that the centers of the pixels lie at the points $(x, y)=(i + \frac{1}{2}, j + \frac{1}{2})$ for both the luminance and chrominance fields. Integer parameters i, j are defined as

- $i = 0, \ldots, H - 1$ and $j = 0, \ldots, V - 1$ for luminance, and
- $i = 0, \ldots, \frac{H}{2} - 1$ and $j = 0, \ldots, \frac{V}{2} - 1$ for chrominance.

This implies that different coordinate systems are used for luminance and chrominance as shown in Figure P.3. Using the coordinate system for chrominance, the integer parameters $u_x^{LT}, u_y^{LT}, u_x^{RT}, u_y^{RT}, u_x^{LB}, u_y^{LB}, u_x^{RB}$, and u_y^{RB} defined above can also be regarded as the x- and y-displacements of the chrominance field at virtual points (x, y)=(0, 0), ($H'/2$, 0), (0, $V'/2$), and ($H'/2$, $V'/2$) in $\frac{1}{64}$-pixel accuracy (the actual displacements are obtained by dividing these values by 64). Using these parameters and a an additional parameter S, which is defined as 2 for luminance and as 1 for chrominance, the resampling algorithm for the luminance and chrominance fields are defined using common equations.

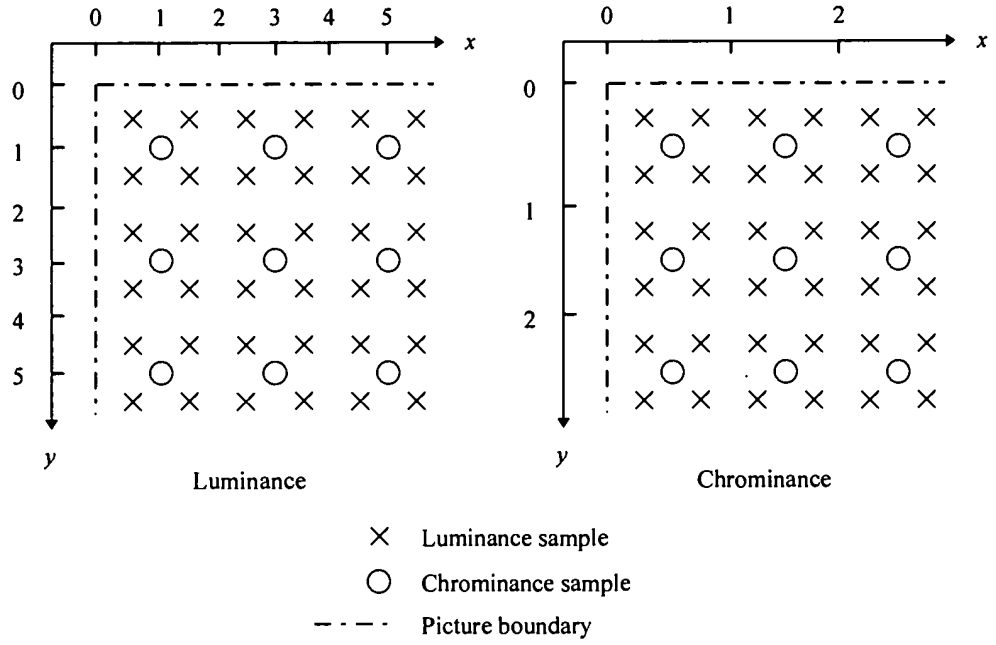


FIGURE P.3/H.263

Coordinate systems for luminance and chrominance fields

The integer parameters $u_x^L(j), u_y^L(j), u_x^R(j)$, and $u_y^R(j)$ which denote the x- and y-displacements of the picture field at (x, y)=$(0, j+\frac{1}{2})$ and $(SH'/2, j+\frac{1}{2})$ in $\frac{S}{64}$-pixel accuracy (the actual displacements are obtained by dividing these values by 64/S) are defined using one dimensional linear interpolation as

$$u_x^L(j) = \left((SV'-2j-1)u_x^{LT} + (2j+1)u_x^{LB}\right) /\!/ (SV') \qquad u_y^L(j) = \left((SV'-2j-1)u_y^{LT} + (2j+1)u_y^{LB}\right) /\!/ (SV')$$

$$u_x^R(j) = \left((SV'-2j-1)u_x^{RT} + (2j+1)u_x^{RB}\right) /\!/ (SV') \qquad u_y^R(j) = \left((SV'-2j-1)u_y^{RT} + (2j+1)u_y^{RB}\right) /\!/ (SV')$$

where "//" denotes integer division that rounds the quotient to the nearest integer, and rounds half integer values away from 0.

Finally, the parameters that specify the transformed position in the reference picture become

$$I_R(i,j) = Pi + \left((SH'-2i-1)u_x^L(j) + (2i+1)u_x^R(j) + 32H'/P\right) /\!/\!/ (64H'/P)$$

$$J_R(i,j) = Pj + \left((SH'-2i-1)u_y^L(j) + (2i+1)u_y^R(j) + 32H'/P\right) /\!/\!/ (64H'/P)$$

$$i_R(i,j) = I_R(i,j) /\!/\!/ P \qquad j_R(i,j) = J_R(i,j) /\!/\!/ P$$

$$\varnothing_x = I_R(i,j) - (I_R(i,j) /\!/\!/ P)P \qquad \varnothing_y = J_R(i,j) - (J_R(i,j) /\!/\!/ P)P$$

where

"///": integer division with truncation towards the negative infinity,

"/": integer division (in this case resulting in no loss of accuracy),

P: accuracy of x- and y-displacements (P=2 when WDA="10" and P=16 when WDA="11" or is absent, see section P.2.1 for the definition of WDA),

$(\frac{I_R(i,j)}{P}+\frac{1}{2},\frac{J_R(i,j)}{P}+\frac{1}{2})$: (x, y) location of the transformed position (both $I_R(i,j)$ and $J_R(i,j)$ are integers),

$(i_R(i,j)+\frac{1}{2}, j_R(i,j)+\frac{1}{2})$: (x, y) location of the sampling point near the transformed position (both $i_R(i,j)$ and $j_R(i,j)$ are integers),

$(\varnothing_x,\varnothing_y)$: bilinear interpolation coefficients of the transformed position (both $\varnothing_x$ and $\varnothing_y$ are integers).

The computation of this equation can be simplified by replacing the divisions by shift operations, since $64H'/P = 2^{m+2}$ when P=16 and $64H'/P = 2^{m+5}$ when P=2.

Using these parameters, the sample value, $E_P(i, j)$, of the pixel located at $(x, y)=(i+\frac{1}{2}, j+\frac{1}{2})$ in the resampled picture is obtained using bilinear interpolation by

$$E_P(i,j) = \Big((P-\varnothing_y)\big((P-\varnothing_x)E_R(i_R,j_R)+\varnothing_x E_R(i_R+1,j_R)\big) + \varnothing_y\big((P-\varnothing_x)E_R(i_R,j_R+1)+\varnothing_x E_R(i_R+1,j_R+1)\big)+P^2/2-1+RCRPR\Big)/P^2$$

where "/" denotes division by truncation. i_R and j_R are simplified notations for $i_R(i, j)$ and $j_R(i, j)$, and $E_R(i_R, j_R)$ denotes the sample value of the pixel located at $(x, y)=(i_R+\frac{1}{2}, j_R+\frac{1}{2})$ in the reference picture after extrapolation using the proper fill mode if necessary. The value of parameter RCRPR is defined as follows:

- For a B-picture (or the B-part of an Improved PB frame) which has a P-picture as its temporally subsequent anchor picture, RCRPR is equal to the rounding type (RTYPE) bit in MPPTYPE (see section 5.1.4.3) of this temporally subsequent P-picture. This implies that for Improved PB frames, RCRPR has the same value for the P-part and the B-part.
- For other types of pictures, RCRPR is equal to the RTYPE bit of the current picture.

P.4 Example of implementation

In this section, an implementation example of the algorithm described in the previous section is provided as pseudo-code.

P.4.1 Displacements of virtual points

When a large picture is coded, a straightforward implementation of the equation for obtaining parameters u_x^{RB} and u_y^{RB} shown in section P.3 may necessitate the usage of variables requiring more that 32 bits for their binary representation. For systems which cannot easily use 64 bit integer or floating point registers, an example of an algorithm which does not require variables with more than 32 bits for the calculation of u_x^{RB} and u_y^{RB} is shown below.

Since H, V, H', and V' are divisible by 4, the definition of u_x^{RB} can be rewritten as

$$u_x^{RB} = \Big((V_Q - V_Q')\big((H_Q - H_Q')u_x^{00} + H_Q' u_x^{H0}\big) + V_Q'\big((H_Q - H_Q')u_x^{0V} + H_Q' u_x^{HV}\big)\Big) // A$$

where $H_Q=H/4$, $V_Q=V/4$, $H_Q'=H'/4$, $V_Q'=V'/4$, $A=H_QV_Q$, and "//" denotes integer division that rounds the quotient to the nearest integer, and rounds half integer values away from 0. Next, parameters T_T and T_B are defined as

$$T_T = (H_Q - H_Q')u_x^{00} + H_Q' u_x^{H0}$$
$$T_B = (H_Q - H_Q')u_x^{0V} + H_Q' u_x^{HV}$$

for simplicity of description. Using operator "///" which denotes integer division with truncation towards the negative infinity, and operator "%" which is defined as $a \% b = a - (a /// b)\, b$, the value of u_x^{RB} can be obtained via the following pseudo-code:

```
q = (VQ - VQ')*(TT ///A) + VQ'*(TB ///A) + ((VQ - VQ')*(TT %A) + VQ'*(TB % A)) ///A;
r = ((VQ - VQ')*(TT %A) + VQ'*(TB % A)) %A;
if (q < 0)
  u_x^RB = q+(r+(A-1)/2)/A;
else
  u_x^RB = q+(r+A/2)/A;
```

The value of u_y^{RB} can also be calculated using this algorithm.

P.4.2 Resampling algorithm

To simplify the description of the algorithm, a function `prior_sample` is defined. Its purpose is to generate a pixel value for any integer location (i_p, j_p) relative to the prior reference picture sampling grid:

```
clip(xmin, x, xmax) {
  if (x < xmin) {
    return xmin;
  } else if (x > xmax) {
    return xmax;
  } else {
    return x;
  }
}

prior_sample(ip, jp) {
  if (FILL_MODE = clip) {
    ic = clip(0, ip, S*HR/2-1);
    jc = clip(0, jp, S*VR/2-1);
    return prior_ref[ic, jc];
  } else {
    if ((ip < 0) OR (ip > S*HR/2-1) OR (jp < 0) OR (jp > S*VR/2-1)) {
      return fill_value;
    } else {
      return prior_ref[ip, jp];
    }
  }
}
```

In the pseudo-code, `prior_ref[i, j]` indicates the sample in column i and row j in the temporally previous reference picture.

Next, a filter function that implements the bilinear interpolation described in section P.3 is defined. It is assumed that all arguments of the following function are integers and that the bilinear interpolation coefficients, $\varnothing_x$ and $\varnothing_y$, are quantized in the range 0,...,P–1 (inclusive).

```
filter(x0, y0, Øx, Øy) {
  return /((P-Øy)·((P-Øx)·prior_sample(x0, y0)+Øx·prior_sample(x0+1, y0))+
           Øy·((P-Øx)·prior_sample(x0, y0+1)+Øx·prior_sample(x0+1, y0+1))+
         P²/2-1+RCRPRJ/P²;.
}
```

Finally, the method for warping the reference picture to generate a prediction for the current picture can be specified in terms of these functions. The pixels of the prediction picture can be generated in a raster scan order. It

is assumed that the values $u_x^L(j), u_y^L(j), u_x^R(j)$, and $u_y^R(j)$ are already calculated and loaded into variables u_x^L, u_y^L, u_x^R, and u_y^R. Defining parameter D as $D = 64H'/P$ and noting that $H'=2^m$, the sample values of the pixels in the j th line of the resampled field (the topmost line is defined as the 0th line) is obtained by the following pseudo-code:

```
a_x^i = D * P + 2 * (u_x^R - u_x^L);
a_y^i = 2 * (u_y^R - u_y^L);
a_x = u_x^L * S * 2^m + (u_x^R - u_x^L) + D / 2;
a_y = j * D * P + u_y^L * S * 2^m + (u_y^R - u_y^L) + D / 2;
for (i = 0; i < S * H / 2; i++) {
    I_R = a_x /// D;
    J_R = a_y /// D;
    i_R = I_R /// P;
    j_R = J_R /// P;
    Ø_x = I_R - (i_R * P);
    Ø_y = J_R - (j_R * P);
    new_ref[i, j] = filter(i_R, j_R, Ø_x, Ø_y);
    a_x += a_x^i;
    a_y += a_y^i;
}
```

where all the variables used in this code are integer variables and `new_ref[i, j]` indicates the sample generated for column i and row j in the resampled reference picture. According to the definition of the parameters, all the divisions in this code can be replaced by binary shift operations. For example, when P=16,

```
I_R = a_x /// D;
J_R = a_y /// D;
i_R = I_R /// P;
j_R = J_R /// P;
Ø_x = I_R - (i_R * P);
Ø_y = J_R - (j_R * P);
```

can be rewritten, assuming that a_x, a_y, I_R, and J_R are binary-coded integer variables in two's complement representation, as

```
I_R = a_x >> (m+2);
J_R = a_y >> (m+2);
i_R = I_R >> 4;
j_R = J_R >> 4;
Ø_x = I_R & 15;
Ø_y = J_R & 15;
```

where "$>> N_{shift}$" denotes a right arithmetic binary shift by N_{shift} bits (N_{shift} is a positive integer), and "&" denotes a bit-wise AND operation.

P.5 Factor-of-4 resampling

Factor-of-4 resampling, which converts both the horizontal and vertical size of the picture by a factor of 2 or $\frac{1}{2}$, is a special case of the resampling algorithm described in section P.3. The simplified description of the resampling algorithm for this special case is provided in this section.

The value of the parameter RCRPR used in the following figures is determined by the rounding type (RTYPE) bit in MPPTYPE (see section 5.1.4.3) as described in section P.3. Additionally, "/" in the following figures indicates division by truncation.

P.5.1 Factor-of-4 upsampling

The pixel value interpolation method used in factor-of-4 upsampling for internal pixels are shown in Figure P.4 By assuming the existence of pixels outside the picture according to the selected fill mode (see sections P.2.3 and P.2.4), the same interpolation method is applied for boundary pixels The interpolation method for boundary pixels when *clip* is selected as the fill mode is shown in Figure P.5. Since precise factor-of-4 upsampling requires x- and y- displacements with at least $\frac{1}{4}$-pixel accuracy, the warping displacement accuracy (WDA) field specified in section P.2.1 must be set to "11" or the resampling must be implicitly invoked in order to use this upsampling method .

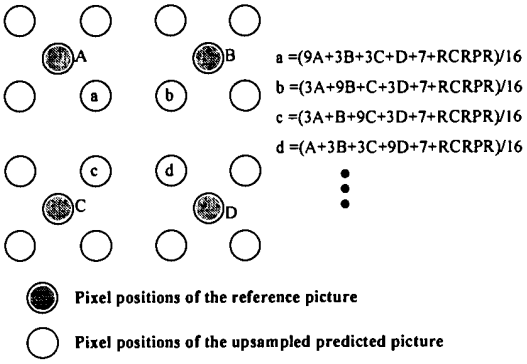


FIGURE P.4/H.263

Factor-of-4 upsampling for pixels inside the picture

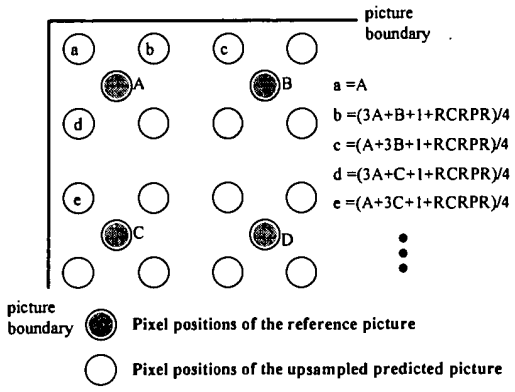


FIGURE P.5/H.263

Factor-of-4 upsampling for pixels at the picture boundary (fill mode = *clip*)

P.5.2 Factor-of-4 downsampling

The pixel value interpolation method for factor-of-4 downsampling is shown in Figure P.6. Since x- and y-displacements with $\frac{1}{2}$-pixel accuracy is sufficient for precise factor-of-4 downsampling, both "10" and "11" are allowed as the value of the displacement accuracy (WDA) field (if present) specified in section P.2.1.

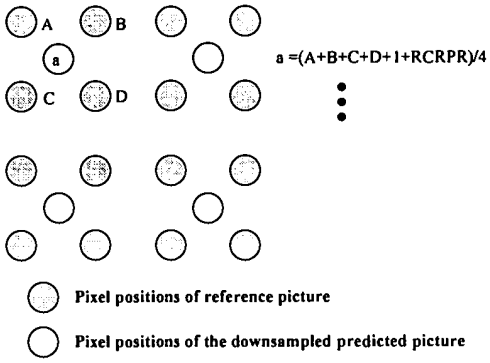


FIGURE P.6/H.263

Factor-of-4 downsampling

Annex Q
Reduced-Resolution Update mode

(This annex forms an integral part of this Recommendation.)

Q.1 Introduction

This annex describes an optional Reduced-Resolution Update mode of H.263. The capability of this mode is signaled by external means (for example Recommendation H.245). The use of this mode is indicated in the PLUSPTYPE field of the picture header.

The Reduced-Resolution Update mode is expected to be used when encoding a highly active scene, and provides the opportunity to increase the coding picture rate while maintaining sufficient subjective quality. This mode allows the encoder to send update information for a picture that is encoded at a reduced resolution, while preserving the detail in a higher resolution reference image to create a final image at the higher resolution.

The syntax of the bitstream in this mode is identical to the syntax for coding without the mode, but the semantics, or interpretation of the bitstream is somewhat different. In this mode, the portion of the picture covered by a macroblock is twice as wide and twice as high. Thus, there is approximately one-quarter the number of macroblocks as there would be without this mode. Motion vector data also refers to blocks of twice the normal height and width, or 32x32 and 16x16 instead of the normal 16x16 and 8x8. On the other hand, the DCT or texture data should be thought of as describing 8x8 blocks on a reduced resolution version of the picture. To produce the final picture, the texture data is decoded at a reduced resolution and then up-sampled to the full resolution of the picture. After upsampling, the full resolution texture image is added to the (already full resolution) motion compensated image to create the image for display and further reference.

In this mode, a picture which has the horizontal size H and vertical size V as indicated in the picture header is created as a final image for display.

In this mode, a referenced picture used for prediction and created for further decoding has a horizontal size H_R and a vertical size V_R which are the same as in the default mode as defined in section 4.1. That is, the H_R and V_R are,

$$H_R = ((H + 15) / 16) * 16$$

$$V_R = ((V + 15) / 16) * 16$$

where H and V are the horizontal size and the vertical size as indicated in the picture header and "/" is defined as division by truncation.

Then in this annex, the texture is coded at a reduced resolution with height and width of H_C and V_C where

$$H_C = ((H_R + 31) / 32) * 32$$

$$V_C = ((V_R + 31) / 32) * 32$$

and "/" is defined as division by truncation.

If H_C and H_R, or V_C and V_R are not identical to each other, such as in the QCIF format, an extension of the referenced picture is performed , and the picture is decoded in the same manner as if the width and height are H_C and V_C. Then the resulting picture which is tiled by 32*32 macroblocks is cropped at the right and the bottom to the width H_R and height V_R, and this cropped picture is stored as a reference picture for further decoding.

If both H and V are the same as those of a resulting picture having the width H_C and the height V_C, this resulting picture is used for display. Otherwise, this resulting picture is further cropped to the size H*V, and the cropped picture here is used for display purpose only.

If the Temporal, SNR, and Spatial Scalability mode (Annex O) or Reference Picture Resampling mode (Annex P) is also used with this option, the source format of the current picture might be different from that of the reference picture. In this case, the resampling of the reference picture shall be performed before decoding.

Note: This mode can be used with the Reference Picture Selection mode (see Annex N) without modification, since the reference picture (after possible resampling by the Reference Picture Resampling mode) has the same size as indicated in the current picture header when this option is used.

Q.2 Decoding Procedure

Figure Q.1 shows the block diagram of the block decoding in the Reduced-Resolution Update mode.

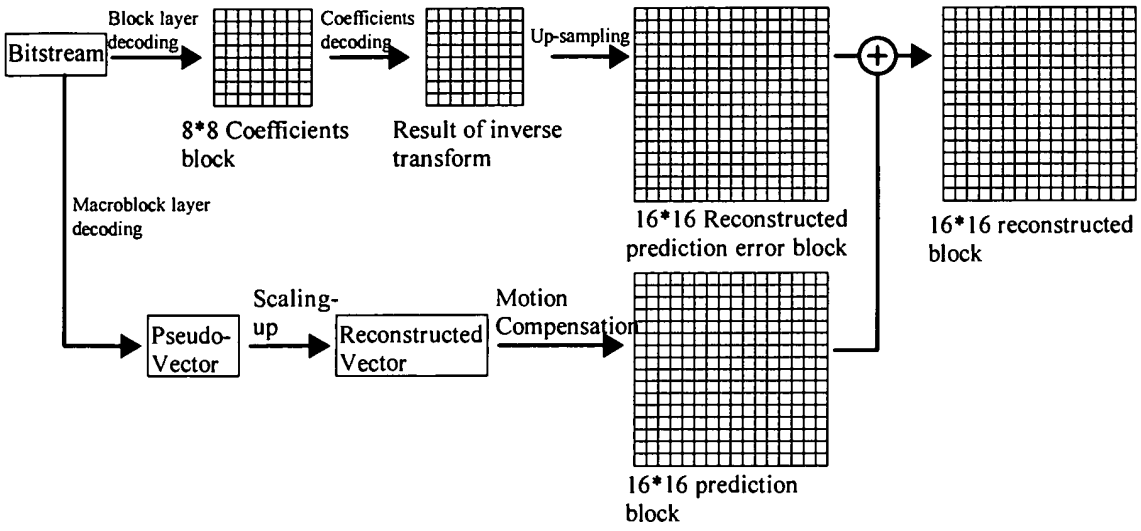


FIGURE Q.1/H.263

Block diagram of block decoding in Reduced-Resolution Update mode

The decoding procedure is given in the following sections:

Q.2.1 Reference Preparation

In some cases, the available reference picture has different size from H_C and V_C. Then the reference picture shall be converted before decoding procedure according to Q.2.1.1 or Q.2.1.2.

Q.2.1.1 Reference Picture Resampling

If the Temporal, SNR, and Spatial Scalability mode (Annex O) or Reference Picture Resampling mode (Annex P) is also used with this option, the source format of the current picture might be different from that of the reference picture. In this case, the resampling of the reference picture according to each Annex shall be performed first.

Q.2.1.2 Extension of Reference Picture

If H_R or V_R is not divisible by 32, such as with the QCIF format, the reference picture is extended. The detailed procedure for this extension is defined in section Q.3.

Q.2.2 Macroblock layer decoding

Decoding can be thought of as operating on "enlarged" blocks of size 32x32 in luminance and 16x16 in chrominance. The texture and motion data for each enlarged block is decoded to create a 32x32 motion block and a 32x32 texture block, as described in Q.2.2.1. and Q.2.2.2. respectively. These motion and texture blocks are then added as described in Q.2.2.3.

Q.2.2.1 Motion Compensation

First, Each component of the macroblock motion vector (or four macroblock motion vectors) is formed from MVD (and possibly MVD_{2-4}). If in Improved PB-Frames mode, MV_F and MV_B for B-Picture is also formed from

MVDB. The detailed procedure for this motion vector formation is defined in section Q.4. If the current picture mode is a B picture or EP-picture, the motion vector for forward and backward are also obtained according to section Q.4.

The motion vector for the two chrominance blocks of the macroblock is obtained from the macroblock motion vector according to section 6.1.1. If either the Advanced Prediction mode or the Deblocking Filter mode is in use and thus four motion vectors are defined for the macroblock, the motion vector for both chrominance blocks is obtained from the four motion vectors according to section F.2. If in the Improved PB-frames mode the creation of the chrominance vector is specified in Annex M. If a B picture or EP-picture is used, the creation of the chrominance vector is specified in Annex O.

Then, a prediction is formed from the motion vector for a INTER Macroblock. Four 16×16 luminance prediction blocks are obtained from the macroblock motion vector, and two 16×16 chrominance prediction blocks are obtained from the chrominance motion vector. For interpolation for subpixel prediction refer to section 6.1.2. If Advanced Prediction mode is also used, an enlarged overlapped motion compensation is performed to obtain four 16×16 luminance prediction blocks using enlarged weighting matrices, for which a detailed procedure is defined in section Q.5. If the current picture mode is an Improved PB-frame, B-picture, or EP-picture, the prediction is obtained according to the other relevant Annexes, except that the size of the predicted blocks is 16×16 instead of 8×8.

Q.2.2.2 Texture decoding

First, the bitstream of the block layer is decoded according to section 5.4. Then coefficients are decoded and the 8×8 reduced-resolution reconstructed prediction error blocks are obtained as the result of inverse transform according to section 6.2.

Then the 16×16 reconstructed prediction error blocks are obtained by upsampling the 8×8 reduced-resolution reconstructed prediction error blocks. For the creation of the edge pixels in each 16×16 reconstructed prediction error block, only the pixels which belong to the corresponding block are used. The detailed procedure is defined in section Q.6.

Q.2.2.3 Reconstruction of block

For each luminance and chrominance block, a summation of the prediction and prediction error is performed. The procedure is identical to section 6.3.1, except that the size of blocks is 16×16 instead of 8×8. Then the clipping is performed according to the section 6.3.2.

Then a block boundary filter is applied to the boundary pixels of the 16×16 reconstructed blocks. The detailed procedure is described in section Q.7.

Q.2. 3 Picture Store

If both H_R and V_R are divisible by 32, such as with the CIF format, the resulting picture reconstructed as described in section Q.2.2 is stored as a reference picture as it is for further decoding. Otherwise, such as the QCIF format, the reconstructed picture which is just covered with 32×32 macroblocks is cropped at the right and the bottom to the width H_R and height V_R, and this cropped picture is stored as a reference picture for further decoding.

Q.2.4 Display

If both H and V are the same as H_C and V_C, the resulting picture in section Q.2.2 is used for display purposes as it is. Otherwise, this resulting picture is further cropped to the size H×V, and the cropped picture is used for display purpose only.

Q.3 Extension of Referenced Picture

If H_R or V_R is not divisible by 32, such as with the QCIF format, the extension of the referenced picture is performed before decoding macroblock/block layer. The width and the height of the extended reference picture for luminance are the next larger size that would be divisible by 32, and those for chrominance are the next larger size that would be divisible by 16.

Note: The width and height of the referenced picture in the default mode are always extended to be divisible by 16 even if picture formats has a width or height that is not divisible by 16, because the picture shall be decoded as if the width or height had the next larger size that would be divisible by 16. See section 4.1/H.263.

If neither Unrestricted Motion Vector mode, Advanced Prediction mode nor Deblocking Filter mode is used with this option, the extended pixels can be arbitrary values, because the extended pixels will be never used as a reference pixels of the decoded picture to be reconstructed and displayed.

If either Unrestricted Motion Vector mode, Advanced Prediction mode or Deblocking Filter mode is also used with this option, the extension of the referenced picture is performed by duplicating the edge pixel of the referenced picture, in order to ensure the decoding when motion vectors point outside the right and bottom edge of the picture.

For example, if the Reduced-Resolution Update mode is used for a QCIF, the width of the referenced picture is 176 and the height is 144, which are not divisible by 32. In order to cover a QCIF picture with 32×32 sized macroblocks, the number of macroblock row should be 6, and the number of macroblock column should be 5. Therefore the width of the extended referenced picture is 192 and the height is 160.

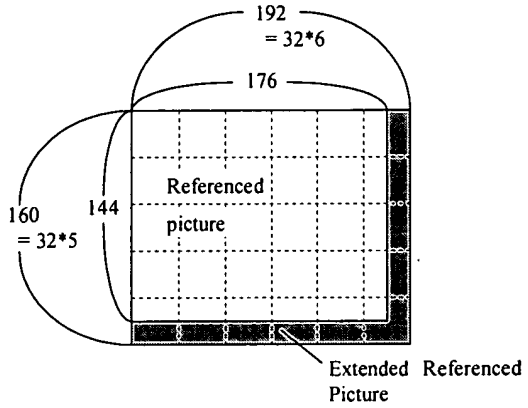


FIGURE Q.2/H.263

Extension of referenced picture for QCIF picture size

The extension of the referenced picture in QCIF is illustrated in Figure Q.2. The extended referenced picture for luminance is given by the following formula:

$R_{RRU}(x, y) = R(x', y')$,
where
x, y = spatial coordinates of the extended referenced picture in the Pixel domain,
x', y' = spatial coordinates of the referenced picture in the pixel domain,
$R_{RRU}(x, y)$ = pixel value of the extended referenced picture at (x, y),
R(x', y') = pixel value of the referenced picture at (x', y'),

x' = 175 if x > 175 and x < 192
= x otherwise,

y' = 143 if y > 143 and y < 160
= y otherwise,

The referenced pictures for chrominance is also extended in the same manner.

Q.4 Reconstruction of motion vectors

In Reduced-Resolution Update mode, the motion vector range is enlarged to approximately double size in both the horizontal and vertical directions. In order to realize an enlarged range using the VLC for MVD defined in Table

14, each vector component is restricted to be only half-pel or zero value. Therefore the range of each motion vector component is [-31.5,30.5] in the default Reduced-Resolution Update mode. If the Unrestricted Motion Vector mode is used, the vector range [-limit,limit] as defined in section D.2 applies to the pseudo-motion vectors and translates into [-(2*limit-0.5),2*limit-1.5] for the motion vectors. For CIF this means that the pseudo-motion vector range is [-32,31.5] and the motion vector range is [-63.5,62.5]. If the UUI field is set to "01", the motion vectors are unlimited. However, the motion vectors (not just the pseudo-motion vectors) are always limited to point not more than 15 pixels outside the coded area as described in section D.1.1. Figure Q.3 illustrates the possible positions of macroblock motion vector or four motion vector predictions around the (0, 0) vector value. The dashed lines indicate the integer coordinates.

For the macroblock using differential motion vectors in a B picture, the motion vector for forward and backward prediction are obtained independently. In the Reduced-Resolution Update mode, the motion vector component MVc for a luminance block is reconstructed from MVD and MVD_{2-4} as follows;

1. Pseudo- prediction vector component **pseudo-P_C** is created from the prediction vector component P_C.

   pseudo-P_C = 0 if P_C = 0

   pseudo-P_C = sign(P_C) * (|P_C| + 0.5) / 2.0 if $P_C \neq 0$

   "/" indicates floating-point division (without loss of accuracy). The prediction vector component P_C is defined as the median value of the vector components MV1, MV2 and MV3 as defined in section 6.1.1/H.263 and F.2/H.263.

2. Pseudo- macroblock vector component **pseudo-MV_C** is obtained by adding the motion vector differences MVD (and MVD_{2-4}) from Table 14 to the pseudo-P_C.

   In the default Reduced-Resolution Update mode, the value of **pseudo-MV_C** is restricted to the range [-16, 15.5]. Only one of the pair will yield a pseudo-MV_C falling within the permitted range. The procedure is performed in the similar way as defined in section 6.1.1/H.263.

   If the Unrestricted Motion Vector mode is also used with the Reduced-Resolution Update mode, **pseudo-MV_C** is obtained by adding the motion vector differences MVD (and MVD_{2-4}) from Table D.3.

   If four motion vectors are present, the procedure is performed in the similar way as defined in section F.2/H.263

3. Motion vector component **MV_C** is obtained from pseudo-MV_C in the following formula:

   $MV_C = 0$ if pseudo-$MV_C = 0$

   MV_C = sign(pseudo-MV_C) * (2.0 * |pseudo-MV_C| - 0.5) if pseudo-$MV_C \neq 0$

   As a result, each vector component is restricted to have a half-integer or zero value, and the range of each motion vector component is enlarged to approximately twice the pseudo-motion vector range.

4. If the current picture mode is an Improved PB frame, or when the MBTYPE indicates the direct mode in a B picture, the motion vector components **MV_F** and/or **MV_B** for forward and backward prediction are created.

   First, pseudo motion vector components **pseudo-MV_F** and/or **pseudo-MV_B** are calculated based on the rules for the prediction modes defined in Annex O or Annex M.

   In the case of the Bidirectional prediction in the Improved PB-frames mode (section M.2.1) or when the MBTYPE indicates the direct mode in a B picture (section O.5.2), pseudo-MV_F and pseudo-MV_B are calculated from **pseudo-MV_D** and **pseudo-MV_C** assuming that pseudo-MV_D is zero and pseudo-MV_C is MV, as defined in Annex G and Annex M.

   In the case of forward prediction in the Improved PB-frames mode (section M.2.2), **pseudo-MV_{DB}** is obtained by decoding the variable length code MVDB according to Table 13. Then pseudo-MV_F is obtained by adding the pseudo-MV_{DB} to the **pseudo-Predictor**. In order to form the pseudo-Predictor, the predictor obtained according to the procedure defined in section M.2.2 of Annex M is converted to the pseudo-Predictor vector according to the formula defined in item 1 of this section.

In the case of Backward prediction in the Improved PB-frames mode (section M.2.3), pseudo-MV_B is set to zero.

Then the motion vectors MV_F and/or MV_B for forward and backward prediction are obtained from pseudo-MV_F and/or pseudo-MV_B according to the formula defined in item 3 of this section.

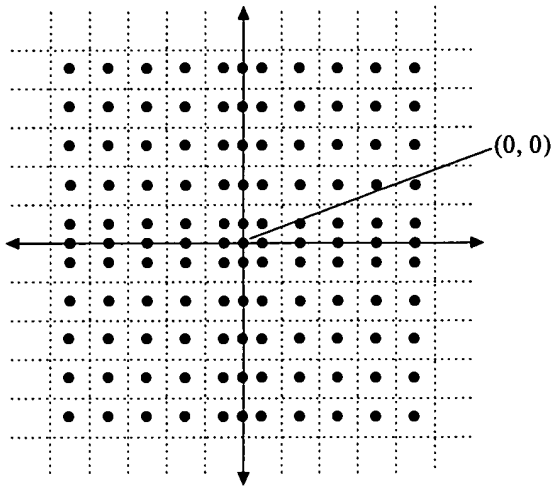


FIGURE Q.3/H.263

Reconstruction of motion vector

Q.5 Enlarged overlapped motion compensation for luminance

If Advanced Prediction mode is also used with Reduced-Resolution Update mode, enlarged matrices of weighting values are used to perform the overlapped motion compensation. Except that the size of each block and weighting matrices is 16×16, the procedure of the creation of each prediction block is identical to the description of F.3 in Annex F.

The enlarged matrices of weighting values for the16×16 luminance prediction are given in Figures Q.4, Q.5, and Q.6.

4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4
4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4
4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4

FIGURE Q.4/H.263

Weighting values, H_0, for prediction with motion vector of current 16×16 luminance block

2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

FIGURE Q.5/H.263

Weighting values, H_1, for prediction with motion vector of current 16×16 luminance blocks on top or bottom of current 16×16 luminance block

2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2

FIGURE Q.6/H.263

Weighting values, H_2, for prediction with motion vector of 16×16 luminance blocks to the left or right of current 16×16 luminance block

Q.6 Upsampling of the reduced-resolution reconstructed prediction error

The 16×16 reconstructed prediction error block is obtained by upsampling the 8×8 reduced-resolution reconstructed prediction error block. In order to realize a simple implementation, filtering is closed within a block which enables to perform an individual upsampling on block basis. Figure Q.7 shows the positioning of samples. The upsampling procedure for the luminance and chrominance pixels which are inside the 16×16 reconstructed prediction error blocks is defined in section Q.6.1. For the creation of the luminance and chrominance pixels which are at the boundary of 16×16 reconstructed prediction error block, the procedure is defined in section Q.6.2. Chrominance blocks as well as luminance blocks are up-sampled. The symbol "/" in Figures Q.8 and Q.9 indicates division by truncation.

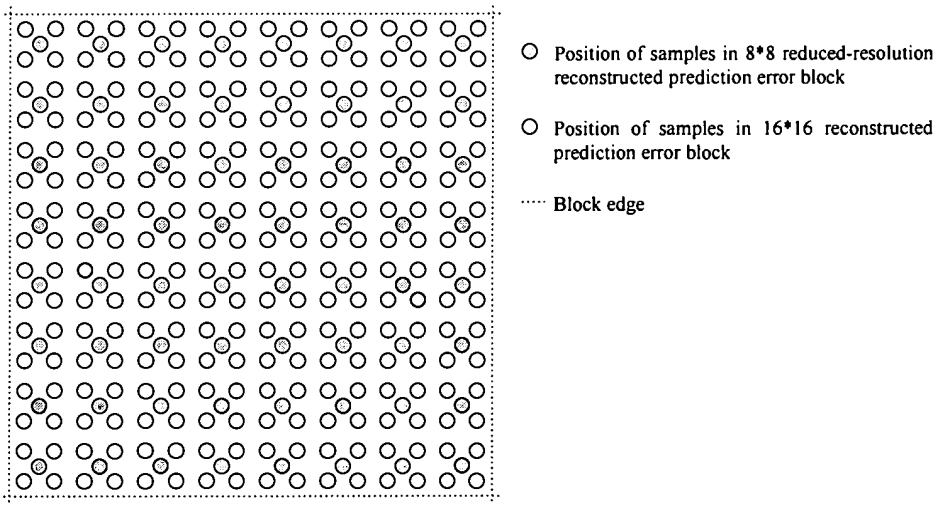


FIGURE Q.7/H.263

Positioning of samples in 8×8 reduced-resolution reconstructed prediction error block and 16×16 reconstructed prediction error block

Q.6.1 Upsampling procedure for the pixels inside a 16x16 reconstructed prediction error block

The creation of reconstructed prediction error for pixels inside block is described in Figure Q.8. "/" indicates division by truncation.

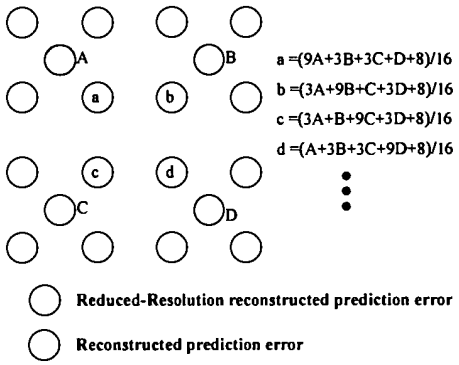


FIGURE Q.8/H.263

Creation of reconstructed prediction error for pixels inside block

Q.6.2 Upsampling procdedure for the pixels at the boundary of 16x16 reconstructed prediction error block

The creation of reconstructed prediction error for pixels of a 16x16 block is shown in Figure Q.9.

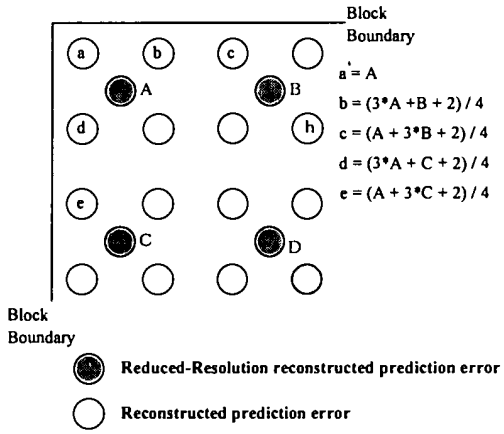


FIGURE Q.9/H.263

Creation of reconstructed prediction error for pixels at the block boundary

Q.7 Block boundary filter

The filter operations are performed along the edges of the 16×16 reconstructed blocks at the encoder as well as on the decoder side. There are two alternative of filtering, depending on whether Deblocking Filter mode is used or not.

The default filtering in Reduced-Resolution Update mode is performed according to section Q.7.1.

If Deblocking Filter mode is also used with Reduced-Resolution Update mode, the filtering is performed according to section Q.7.2.

In both cases, filtering is performed on the complete reconstructed image data before storing the data in the picture store for future prediction. No filtering is performed across picture edges, slice edges in Slice Structured mode (see Annex K), or GOB boundaries having GOB headers present in Independent Segment Decoding mode (see Annex R). Chrominance as well as luminance data is filtered.

Q.7.1 Definition of the default block boundary filter

In the Reduced-Resolution Update mode, the default filtering is performed according to this section.

If A and B are two pixel values on a line — horizontal or vertical — of the reconstructed picture, and A belongs to one 16×16 block called block1 whereas B belongs to a neighboring 16×16 block called block2 which is to the right or below of block1. Figure Q.10 shows examples for the position of these pixels.

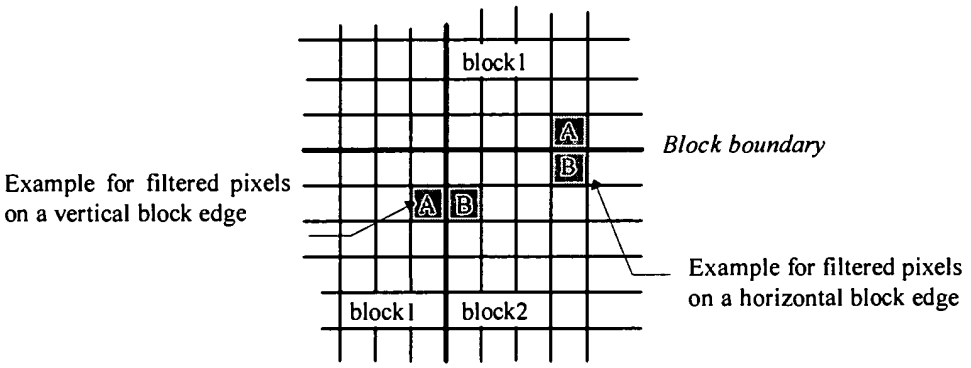


FIGURE Q.10/H.263

Default block boundary filter

One of the following conditions must be fulfilled in order to turn the filter on for a particular edge:

- block1 belongs to a coded macroblock (COD==0 || MB-type == INTRA) **or**
- block2 belongs to a coded macroblock (COD==0 || MB-type == INTRA)

A shall be replaced by A1 and B shall be replaced by B1. "/" indicates division by truncation.

A1 = (3*A + B + 2) / 4

B1 = (A + 3*B + 2) / 4

The order of edges where filtering is performed is identical to the description provided in section J.3 of Annex J.

Q.7.2 Definition of the block boundary filter when Deblocking Filter mode is used

If the Deblocking Filter mode (see Annex J) is used with the Reduced-Resolution Update mode, the filtering which is defined in Annex J with one modification is performed on the boundary pixels of 16×16 luminance and chrominance blocks, in place of the filtering described in section Q.7.1. The one modification of the filtering in Annex J is that the parameter STRENGTH is given the value of positive infinity. This implies that the function UpDownRamp(x, STRENGTH) defined in section J.3 becomes a linear function of x.

As a result, the procedure of the deblocking filter described in section J.3 is redefined in the following manner;

B1 = clip(B + d1)
C1 = clip(C – d1)
A1 = A – d2
D1 = D + d2
d1 = (A–4B+4C–D) / 8
d2 = clipd1((A – D) / 4, d1/2)

Annex R
Independent Segment Decoding mode

(This annex forms an integral part of this Recommendation.)

R.1 Introduction

This annex describes the optional Independent Segment Decoding mode of H.263, which allows a picture to be decoded without the presence of any data dependencies across slice boundaries or across GOB boundaries having non-empty GOB headers. The use of this mode is indicated in the PLUSPTYPE field of the picture header. The capability to use this optional mode is negotiated by external means (for example Recommendation H.245).

When the use of this mode is indicated, the video picture segment boundaries (as defined by the boundaries of the slices or the upper boundaries of the GOBs for which GOB headers are sent, or the boundaries of the picture, whichever bounds a region in the smallest way) are treated as picture boundaries when decoding, including the treatment of motion vectors which cross those boundaries (which result in boundary extrapolation when the Unrestricted Motion Vector mode, the Advanced Prediction mode, the Deblocking Filter mode, or the Temporal, SNR, and Spatial Scalability mode are in use, and which are prohibited when none of those optional modes are in use).

R.2 Mode Operation

A video picture segment is defined by the following:

If the Slice Structured mode (see Annex K) is not in use, then one GOB or a plural number of consecutive GOBs forms one video picture segment. The location of the top of each video picture segment is indicated by the presence of a non-empty GOB header for which the border of the video picture segment lies just above the macroblocks in the GOB for which a header is present, or the top of the picture, whichever is lower. The location of the bottom of each video picture segment is defined by the top of the next video picture segment, or the bottom of the picture, whichever is uppermost.

If the Slice Structured mode (see Annex K) is in use, then each slice forms one video picture segment.

In the Independent Segment Decoding mode, each video picture segment is decoded with complete independence from all other video picture segments, and is also independent of all data outside the same video picture segment location in the reference picture(s). This includes:

1. No use of motion vectors outside of the current video picture segment for motion vector prediction (as in section 6.1.1)
2. No use of motion vectors outside of the current video picture segment as remote motion vectors for overlapped block motion compensation when the Advanced Prediction mode is in use (see section F.3)
3. No deblocking filter operation across video picture segment boundaries (see section J.3)
4. No use of motion vectors which reference data outside the current video picture segment unless the Unrestricted Motion Vector mode (see Annex D), the Advanced Prediction mode (see Annex F), the Deblocking Filter mode (see Annex J) or the Temporal, SNR, and Spatial Scalability mode (see Annex O) are in use; in which case the borders of the current video picture segment in the prior picture are extrapolated as described in Annex D to form predictions of the pixels which reference the out-of-bounds region.
5. No bilinear interpolation across the boundaries of the ¼-size or ½-size region corresponding to the current video picture segment for upward prediction in spatial scalability EI and EP pictures (as defined in Annex O).
6. When the Reduced Resolution Update mode (see Annex Q) is in use, no block boundary filter operation across video picture segment boundaries.
7. No use of the Reference Picture Resampling mode with the Independent Segment Decoding mode.

R.3 Constraints on Usage

Certain restrictions are placed on the use of other aspects of the video coding syntax when the Independent Segment Decoding mode is in use. These restrictions are made to prevent two pathological cases which otherwise would make operation of the Independent Segment Decoding mode difficult.

R.3.1 Constraint on segment shapes

In the use of the Slice Structured mode (Annex K) without the use of the Rectangular Slice submode (section K.1), there can arise cases in which the shape of a video picture segment may be non-convex (having "inside corners", or even comprising two distinct and separated regions of the picture).

Therefore, the Independent Segment Decoding mode shall not be used with the Slice Structured mode without the simultaneous use of the Rectangular Slice submode of the Slice Structured mode (see Annex K). This constraint is mandated to prevent the need for difficult special-case treatment in order to determine how and when to perform extrapolation of each video picture segment.

R.3.2 Constraint on changes of segment shapes

If the shape of the video picture segments were allowed to change in any way from picture to picture in the bitstream, there could arise cases in which the bitstream would be difficult to decode. This is because in such cases the bitstream content is not sufficient to determine the shape of each video picture segment prior to the possible appearance of motion vectors in the bitstream which require knowledge of the video picture segment shape for proper interpretation.

Therefore, when the Independent Segment Decoding mode is in use, the video picture segmentation for all pictures and frames allowing temporal prediction (i.e., all P, B, and EP pictures and all Improved PB frames) shall be the same as that used in its temporal reference picture. Also, when the Independent Segment Decoding mode is in use, the video picture segmentation for all EI pictures shall either be the same or shall differ only by sub-dividing the video picture segmentation used in its reference picture. Also, the Independent Segment Decoding mode shall not be used in any picture or frame which uses reference pictures (all picture types except INTRA) unless the Independent Segment Decoding mode is also used in all of the reference picture(s) for the current picture. As a result of this constraint, the shape of the video picture segments in the Independent Segment Decoding mode shall never change from picture to picture except as changed in I and EI pictures (and the manner in which EI pictures can change segmentation is itself also somewhat constrained).

Annex S
Alternative INTER VLC mode

(This annex forms an integral part of this Recommendation.)

S.1 Introduction

This annex describes an optional Alternative INTER VLC mode of H.263, which improves the efficiency of inter-picture coding when significant changes are evident in the picture. This efficiency improvement is obtained by allowing some VLC codes originally designed for INTRA pictures to be used for some INTER picture coefficients and CBPY data as well. The use of this mode is indicated in the PLUSPTYPE field of the picture header. The capability to use this optional mode is negotiated by external means (for example Recommendation H.245). The mode contains two syntax alterations, one for the encoding of INTER coefficients and another for the encoding of the INTER CBPY values.

S.2 Alternative INTER VLC for coefficients

The concept behind the design of the INTRA VLC table of Annex I is to use the same codewords as in the original INTER VLC but with a different interpretation of LEVEL and RUN. The INTRA VLC is better suited in cases where there are many and/or large-valued coefficients.

The INTRA VLC is constructed so that codewords have the same value for LAST (0 or 1) in both the INTER and INTRA tables. The INTRA table is therefore produced by "reshuffling" the meaning of the codewords with the same value of LAST. Furthermore, for events with large |LEVEL| the INTRA table uses a codeword which in the INTER table has a large RUN. In INTER blocks having a large number of large-magnitude coefficients, it can sometimes be more efficient to use the INTRA table than the INTER table, and in some such cases the choice of the VLC table can be apparent to the decoder since decoding using the INTER table would result in RUN values so large as to indicate the presence of more than 64 coefficients for a block. Under these circumstances, the INTRA table can be used to improve the efficiency of INTER coding.

S.2.1 Encoder action.

The encoder may use the INTRA VLC table for coding an INTER block whenever the decoder can detect its use—in other words, whenever decoding using the INTER VLC table would cause coefficients outside the 64 coefficients of a block to be addressed.

The encoder would normally choose to use the INTRA VLC table for coding an INTER block only when the above condition is satisfied and also when the INTRA VLC usage results in fewer bits than the INTER VLC for the same coefficient values. This will often be the case when there are many large coefficients, due to the way the INTRA VLC was produced (since the ordinary INTER VLC table contains long run-lengths for the same codewords in which the INTRA VLC table contains large coefficient amplitudes).

S.2.2 Decoder action

The decoding process is as follows:

1. The decoder first receives all coefficient codes of a block.
2. The codewords are then interpreted assuming that INTER VLC is used. If the addressing of coefficients stays inside the 64 coefficients of a block, the VLC decoding is finished.
3. If coefficients outside the block are addressed, the codewords shall be interpreted according to the INTRA VLC.

S.3 Alternative INTER VLC for CBPY

The INTER CBPY codewords (Table 13) are designed with the assumption that there are more Y blocks with all zero coefficients than there are with at least one non-zero coefficient. When both C_B and C_R blocks have at least one non-zero coefficient, i.e., $CBPC_5 = CBPC_6 = 1$, this assumption no longer holds. For this reason, when the Alternative INTER VLC mode is in use, the CBPY codewords as defined in Table 13 for INTRA macroblocks shall also be used for INTER macroblocks whenever $CBPC_5 = CBPC_6 = 1$.

Annex T
Modified Quantization mode

(This annex forms an integral part of this Recommendation.)

T.1 Introduction

This annex describes an optional Modified Quantization mode of H.263, which modifies quantizer operation. The use of this mode is indicated in the PLUSPTYPE field of the picture header. The capability to use this optional mode is negotiated by external means (for example Recommendation H.245).

This mode includes four key features:

1. The bit-rate control ability for encoding is improved by altering the syntax for the DQUANT field,
2. Chrominance fidelity is improved by specifying a smaller step size for chrominance than that for luminance data,
3. The range of representable coefficient values is extended to allow the representation of any possible true coefficient value to within the accuracy allowed by the quantization step size,
4. The range of quantized coefficient levels is restricted to those which can reasonably occur, to improve the detectability of errors and minimize decoding complexity.

T.2 Modified DQUANT Update

This mode modifies the semantics of the DQUANT field. With this mode, it is possible to use DQUANT either to modify QUANT by plus or minus a small amount, or to signal any specific new value for QUANT. The size of the small amount of modification depends on the current value of QUANT. By use of this mode, more flexible control of the quantizer step size can be specified in the DQUANT field.

The codeword for DQUANT in this mode is no longer a two bit fixed length field. It is a variable length field which can either be two bits or six bits in length. Whether it is two or six bits depends on the first bit of the code. The description below is therefore split into two sections, depending on the first bit.

T.2.1 Small-step QUANT alteration

When the first bit of the DQUANT field is 1, only one additional bit is sent in DQUANT. The single additional bit is used to modify QUANT by a differential value. The change in the value of QUANT is dependent on the second bit of DQUANT and on the prior value of QUANT, as shown in Table T.1.

Example: If the previous value of QUANT is 29 and DQUANT is signalled with the codeword '11', then the differential value is +2, and thus the resulting new QUANT value is 31.

TABLE T.1/H.263

Semantics of small-step QUANT alteration

	Change of QUANT	
Prior QUANT	DQUANT = 10	DQUANT = 11
1	+2	+1
2–10	–1	+1
11–20	–2	+2
21–28	–3	+3
29	–3	+2
30	–3	+1
31	–3	–5

T.2.2 Arbitrary QUANT selection

When the first bit of the DQUANT field is 0, five additional bits are sent in DQUANT. The following five bits represent a new QUANT as defined in section 5.1.19.

Example: Regardless of the current value of QUANT, if DQUANT is signaled with the code word '001111', then the new value of QUANT is 15.

T.3 Altered quantization step size for chrominance coefficients

When the modified quantization mode is in use, the quantization parameter of the chrominance coefficients is different from the quantization parameter of the luminance. The luminance quantization parameter is signaled in the bitstream. It is called QUANT. When this mode is in use, a different quantization parameter termed QUANT_C is used for the inverse quantization of chrominance coefficients. The relation between QUANT and QUANT_C is given in Table T.2. If the deblocking filter mode (see Annex J) is in use, QUANT_C shall also be used for the application of the deblocking filter to the chrominance data. Whenever QUANT is discussed herein in any other context, it shall mean the quantization step size of the luminance.

TABLE T.2/H.263

Relationship between QUANT and QUANT_C

Range of QUANT	Value of QUANT_C
1–6	QUANT_C = QUANT
7–9	QUANT_C = QUANT – 1
10–11	9
12–13	10
14–15	11
16–18	12
19–21	13
22–26	14
27–31	15

T.4 Modified coefficient range

When the Modified Quantization mode is in use, quantized DCT coefficients having quantization level magnitudes greater than 127 can be represented in the bitstream. This has two advantages:

1. Encoder performance is improved by allowing the true full range of possible coefficient values to be represented, and
2. Encoder complexity is reduced by eliminating the need to increase the quantization step size upon encountering certain large coefficient values which would otherwise be unrepresentable.

It is possible that the correct value of a DCT coefficient prior to quantization in the encoder will have a magnitude as high as 2040. Thus, a range of –127 to +127 for LEVEL is insufficient to cover the entire range of possible coefficient values whenever the quantization parameter QUANT or QUANT_C is less than 8. The expanded coefficient range broadens the range of LEVEL to allow any true coefficient value to be more properly encoded.

When the Modified Quantization mode is in use, the meaning of the LEVEL field following an ESCAPE code (0000 011, as per section 5.4.2) is altered. In this mode, rather than being forbidden, the bit sequence 1000 0000 is used to represent an EXTENDED-ESCAPE code. An AC coefficient of magnitude greater than 127 is represented by sending an EXTENDED-ESCAPE code, immediately followed by a fixed-length EXTENDED-LEVEL field of 11 bits. An extended coefficient value is encoded into the EXTENDED-LEVEL field by taking the least significant 11 bits of the two's-complement binary representation of LEVEL and cyclically rotating them to the right by 5 bit

positions. This rotation is necessary to prevent start code emulation. The cyclic rotation is illustrated in Figure T.1.

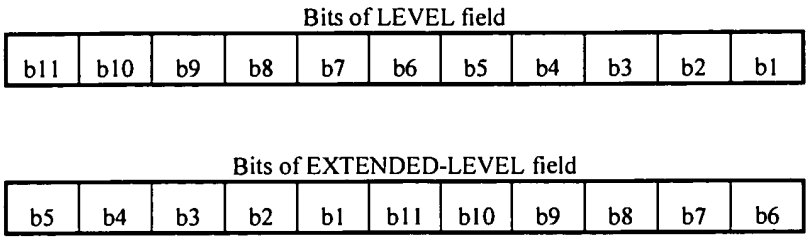


FIGURE T.1/H.263

Cyclic rotation of coefficient representation

T.5 Usage Restrictions

When the Modified Quantization mode is in use, certain restrictions are placed on the encoded coefficient values. This has several benefits:

1. The detectability of bit errors is improved by prohibiting certain unreasonable coefficient values, thus allowing these values to be recognized as bit errors by the decoder; and
2. Decoder complexity is reduced by reducing the wordlength necessary for inverse quantization prior to clipping.
3. Start-code emulation is prevented for coefficients encoded using the EXTENDED-ESCAPE mechanism described in section T.4.

These restrictions are as follows. When the Modified Quantization mode is in use:

1. For any coefficient, the reconstruction level magnitude |REC| produced by the inverse quantization process described in section 6.2.1 using the current value of QUANT or QUANT_C as appropriate, and the encoded value of LEVEL, shall be less than 4096. This additional restriction applies to all coefficients, regardless of whether the coefficient is sent using the EXTENDED-ESCAPE mechanism or not.
2. The bitstream shall not use either the normal ESCAPE code or the EXTENDED-ESCAPE code to encode a combination of LAST, RUN, and LEVEL for which there exists a codeword entry in the applicable VLC table, which is either Table 16 (section 5.4.2) or Table I.2 (section I.3).
3. The EXTENDED-ESCAPE code shall be used only when the quantization parameter for the coefficient (QUANT or QUANT_C) is less than eight (8).
4. The EXTENDED-ESCAPE code shall be used only when it is followed by an EXTENDED-LEVEL field representing a value of LEVEL which is outside of the range –127 to +127.

Appendix I
List of patent holders

(This appendix does not form an integral part of this Recommendation)

[Ed. Note: Appendix I should be kept the same as in the existing published Recommendation H.263 Version 1, provided it exists, and should be updated as appropriate. If Appendix I does not exist, then the content shown below as Appendix III should be adopted as Appendix I.]

Appendix II
Error Tracking

(This appendix does not form an integral part of this Recommendation)

[Ed. Note: Appendix II should be kept the same as in the existing published Recommendation H.263 Version 1. (It does exist, but it is not found in the published Recommendation on the ITU site as of the time of the preparation of this draft.)]

Appendix III
Recommended Optional Enhancement

(This appendix does not form an integral part of this Recommendation)

III.1 Introduction

With the variety of optional modes available in this Recommendation (H.263 Version 2), it is crucial that several preferred mode combinations for operation be defined, so that option-enhanced terminals will have a high probability of connecting to each other using some syntax better than the "baseline." This appendix contains a list of preferred mode combinations, structured into three "levels" of support. Each level includes support for the levels below it, creating an "onion peel" structure of capabilities. The primary objective of this appendix is to describe the order in which modes should be supported in decoders, rather than to enforce a particular small set of mode combinations upon encoders.

In determining which modes would be placed into this level structure, the primary criteria used were performance-related: improvement in subjective quality, impact on delay, and impact on complexity (this includes computational burden, data dependencies, and ease of implementation). However, because this appendix seeks to address a wide variety of applications and transport layers, features addressing error resilience and ease of packetization were also incorporated into the same level structure. In keeping with these objectives, the levels of support described in this appendix are intended to be *universal*, so that the means of transport will not be an issue and the need for application-specific "profiles" will be minimized. This universality is seen as a key to promoting good interoperability across different types of terminals and networks. To maintain universality, several optional features have not been included in the level structure described herein. The absence of these features is not meant to be a condemnation of their usage; instead, their absence reflects only a collective opinion about the likelihood of widespread adoption of these features across a full spectrum of terminals, networks, and applications.

III.2 Levels of Preferred Mode Support

Each optional mode described in this appendix has been categorized into one of three levels of support. While the ratio of performance improvement to computational complexity was a major criterion used in this categorization, utility for packetization and error resilience were also considered. Decoder support for a given level implies support for all lower levels. Furthermore, decoder support for a given level implies support for all subset combinations of the constituent modes of that level and all lower levels. This last requirement exists so that the limitations placed upon an encoder's choice of mode combinations is minimized. This is in keeping with the primary objective of this appendix, which is to describe the order in which modes should be supported at the decoder, rather than to enforce a particular small set of mode combinations upon the encoder.

Backward compatibility is also an important issue to consider. For this reason, a preferred mode combination including only the Advanced Prediction mode (Annex F) is also recommended (although this recommendation falls outside the level structure defined below), since Advanced Prediction is regarded as the most beneficial of the optional modes in the original edition (Version 1) of this Recommendation.

III.2.1 Level 1 Preferred Modes

The first level of support is composed of the following modes:

1. **Advanced INTRA Coding (Annex I).** Use of this mode improves the coding efficiency for INTRA macroblocks (whether within INTRA pictures or predictively-coded pictures). The additional computational requirements of this mode are minimal at both the encoder and decoder (as low as a maximum of 8 additions/subtractions per 8×8 block in the decoding process plus the use of a different but very similar VLC table in order to obtain a significant improvement in coding efficiency). For these reasons, Advanced INTRA Coding is included at this lowest level of support.

2. **Deblocking Filter (Annex J).** Because of the significant subjective quality improvement that may be realized with a deblocking filter, these filters are already widely in use as a method of post-processing in video communication terminals. Annex J represents the preferred mode of operation for a deblocking filter because it places the filter within the coding loop. This placement eases the implementation of the filter (by reducing the memory requirement) and somewhat improves the coding performance over a post-procesing implementation. As with the Advanced Prediction mode, this mode also includes the four-motion-vector-per-macroblock feature and picture boundary extrapolation for motion compensation, both of which can further improve coding efficiency. The computational requirements of the deblocking filter are several hundred

operatons per coded macroblock, but memory accesses and computational dependencies are uncomplicated. This last point is what makes the Deblocking Filter preferable to Advanced Prediction. Also, the benefits of Advanced Prediction are not as substantial when the Deblocking Filter is used as well. Thus, the Deblocking Filter is included at this lowest level of support while Advanced Prediction is deferred to Level 3.

3. **Supplemental Enhancement Information (Full-Frame Freeze Only) (Annex L Section L.4).** The full-frame freeze is very simple to implement, requiring only that the decoder be able to stop the transfer of data from its output buffer to the video display. This sub-mode is useful for preventing the display of low-fidelity pictures while the encoder is building up a higher fidelity picture.

4. **Modified Quantization (Annex T).** This mode includes an extended DCT coefficient range, modified DQUANT syntax, and a modified step size for chrominance. The first two features allow for more flexibility at the encoder and may actually decrease the encoder's computational load (by eliminating the need re-encode macroblocks when coefficient level saturation occurs). The third feature noticeably improves chrominance fidelity, typically with little added bit-rate cost and with virtually no increase in computation. At the decoder, the only significant computational burden is the ability to parse several new bitstream symbols.

III.2.2 Level 2 Preferred Modes

The second level of support is composed of the following modes:

1. **Unrestricted Motion Vectors (With UUI = "1" Sufficient) (Annex D).** Annex D has two primary features: (1) picture boundary extrapolation, and (2) longer motion vector support. The first is already supported by the inclusion of Annex J in the first level of support. The longer motion vector support can provide a significant improvement in coding efficiency, especially for large picture sizes, rapid motion, camera movement, and low picture rates. When used with PLUSPTYPE present, this mode also allows for longer motion vector differences, which can significantly simplify encoder operation. The longer motion vectors do present a potential problem for the decoder in terms of memory access, but frame-size-dependent limits on the maximum motion vector size prevent this problem from becoming an appreciable obstacle to implementation.

2. **Slice Structured Mode (Annex K).** For compatibility with packet-based transport layers, the Slice Structured mode is included here. All submodes of Annex K are to be supported, including the Rectangular Slice submode and the Arbitrary Slice Ordering submode. The additional computational burden imposed by the Slice Structured mode is minimal, limited primarily to bitstream generation and parsing.

3. **Reference Picture Resampling (Implicit Factor-of-4 Mode Only) (Annex P).** The implicit factor-of-4 mode of Reference Picture Resampling allows for automatic reference picture resampling only when the size of the new frame is changed, as indicated in the picture header. No bitstream overhead is required for this mode of operation. Predictive dynamic resolution changes allow an encoder to make intelligent trade-offs between temporal and spatial resolution. Furthermore, this simplest mode of operation for Annex P (factor of 4 upsampling or downsampling only) adds only a modest amount of computational complexity to both the encoder or decoder, since the factor of 4 case uses a simple fixed FIR filter (requiring roughly 4 operations per pixel, at most).

III.2.3 Level 3 Preferred Modes

The third and final level of support is composed of the following modes:

1. **Advanced Prediction (Annex F).** From a coding efficiency standpoint, this mode is the most important of the modes available in the prior version (Version 1) of this Recommendation. It includes overlapped block motion compensation, the four-motion-vector-per-macroblock feature, and it allows for motion vectors to point outside of the picture boundaries. The use of Advanced Prediction results in significant improvements in both subjective and objective performance. It does, however, require an appreciable increase in computation. More importantly, the data dependencies introduced by Advanced Prediction result in a complicated order of processing at the decoder. Coupled with the fact that the quality improvements are not as obvious when the Deblocking Filter is also in use, these complexity considerations are the reason this mode is part of the third support level, rather than the first. However, since implementations of Recommendation H.263 that were designed prior to the adoption of the other modes in this list might have implemented Advanced Prediction by itself, Advanced Prediction-only operation is also recommended for maximal backward compatibility.

2. **Improved PB-frames (Annex M).** The Improved PB-frames mode represents a substantial improvement over the basic PB-frames mode (Annex G) and has been demonstrated to significantly improve the coding efficiency for video sequences with a low degree of motion. While the basic version of PB-frames had problems in handling scene cuts and irregular motion, the Improved PB-frames mode may be used effectively at all times. In terms of computational burden, this mode is roughly neutral (especially for macroblock-pipelined architectures). However, because the Improved PB-frames mode can have a negative impact on delay and

because it requires an additional frame store to implement (at both the encoder and decoder), it is placed in a higher layer.

3. **Independent Segment Decoding (Annex R).** While the Independent Segment Decoding mode does introduce a significant amount of complexity (in terms of ease of implementation), it also significantly improves the error robustness of a video decoder, particularly in the presence of packet losses. For this reason, it is likely that many applications in which data can be lost in transmission (whether through packet loss or channel interference) will make use of the Independent Segment Decoding mode, especially in combination with the Slice Structured mode.

4. **Alternate INTER VLC (Annex S).** This mode has been demonstrated to improve coding efficiency especially for pictures with heavy motion which are coded with high fidelity. This mode uses the same VLC table as the Advanced INTRA Coding mode and is also simple from a computational standpoint (for each block, it requires at most one additional pass over the parsed symbols to decode them into DCT coefficients). However, it is possible that this mode could lead to some difficulty in implementation (especially on highly pipelined architectures), which is why it is not included at the lower levels of support.

III.3 Picture Formats and Picture Clock Frequencies

To ensure a high quality level of interoperability, encoders and decoders supporting a large standard picture format (QCIF, CIF, 4CIF, 16CIF) should support all smaller standard picture formats. (As specified elsewhere in this Recommendation, decoders shall support Sub-QCIF and QCIF, and encoders shall support Sub-QCIF or QCIF.) Decoders should be capable of operation with a smaller picture format at maximum frame rates no lower than the maximum frame rate for which it is capable of operation with a larger standard picture format. For example, a decoder capable of decoding 4CIF pictures at 15 pictures per second (actually 15*(1000/1001) pictures per second) should also be able to decode CIF, QCIF and SQCIF pictures at least at 15 pictures per second.

Encoders and decoders supporting custom picture formats and/or custom picture clock frequencies are recommended to follow the guidelines below:

1. An encoder or decoder which supports a custom picture format should support all standard picture formats equal or smaller in both height and width than those of any supported custom picture format. For example, an encoder supporting a custom picture format of 256×256 should support QCIF and SQCIF picture encoding.

2. An encoder or decoder which supports a custom picture clock frequency for a standard picture format should support a higher or equal frame rate (in pictures per second) at the standard picture clock frequency of 30 Hz (actually 30*(1000/1001) Hz) for the same picture format. For example, if a decoder supports QCIF pictures at 12.5 pictures per second with a 25 Hz picture clock, it should at least be able to decode QCIF pictures at 15 pictures per second at 30 Hz. If a custom picture format is used, all supported smaller or equal standard picture formats should be capable of operating at frame rates for the standard picture clock frequency no lower than the maximum frame rate of the custom picture format.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☑ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

ITU - Telecommunications Standardization Sector
STUDY GROUP 16
Video Coding Experts Group (VCEG)

Document VCEG-N10
Filename: VCEG-N10.doc
Generated: 7/10/01

Question: Q.6/SG16(VCEG)

Source: Gisle Bjontegaard (editor)
Telenor Broadband Services
P.O.Box 6914 St.Olavs plass
N-0130 Oslo, Norway

Tel: +47 23 13 83 81
Fax: +47 22 77 79 80
Email: gisle.bjontegaard@telenor.com

With TML-8 Changes edited by Thomas Wiegand

Title: H.26L Test Model Long Term Number 8 (TML-8) draft0.

Purpose: Test model

This d0 version of TML-8 includes the following changes from TML-6 (adopted in Austin and Porto Seguro):

- Run coding of coded MBs M29,M57
- 1/8 pixel prediction accuracy M45
- DQUANT on MB level M31
- Vectors pointing outside picture M34
- Dropping RDquant M72
- Drop of isolated chroma AC coefficients M32
- New sections7,8 for data partitioning and NAL(input not received yet) M52
- Some bug fixes
- New section for entropy coding including the description of UVLC and CABAC. (D. Marpe)
- Section "Motion Estimation and Mode Decision" with new High-complexity mode (H. Schwarz)

1 SCOPE 5

2 ORGANIZATION OF SOURCE- AND COMPRESSED DATA 5

2.1 Picture formats 5

2.2 Subdivision of a picture into macroblocks 5

2.3 Order of the bitstream within a macroblock 6

2.4 Syntax 7
2.4.1 Syntax diagram 7

3 DESCRIPTION OF SYNTAX ELEMENTS 8

3.1 Picture sync 8

3.2 Picture type (Ptype) 8

3.3 RUN 8

3.4 Macro block type (MB_Type) 8
3.4.1 Intra 8
3.4.2 Inter 9

3.5 Intra prediction mode (Intra_pred_mode) 9
3.5.1 Mode 0: DC prediction 9
3.5.2 Mode 1: 9
3.5.3 Mode 2: Vertical prediction 10
3.5.4 Mode 3: Diagonal prediction 10
3.5.5 Mode 4:Horizontal prediction 10
3.5.6 Mode 5: 10
3.5.7 Prediction of chroma blocks 10
3.5.8 Coding of Intra prediction modes 11
3.5.9 Intra mode based on 16x16 macroblocks (16x16 intra mode) 11
3.5.9.1 Prediction modes 11
3.5.9.2 Residual coding 12
3.5.9.3 Signalling of mode information for 16x16 intra coding 12

3.6 Reference frame (Ref_frame) 12

3.7 Motion Vector Data (MVD) 13
3.7.1 Fractional pixel accuracy 13
3.7.1.1 1/4 pixel accuracy 13
3.7.1.2 1/8 pixel accuracy 14
3.7.2 Prediction of vector components 14
3.7.2.1 Median prediction 14
3.7.2.2 Directional segmentation prediction 15
3.7.3 Chroma vectors 15

3.8 Coded Block Pattern (CBP) 16

3.9 Dquant 16

4 TRANSFORM AND INVERSE TRANSFORM 16

4.1 4x4 block size 16

4.2 2x2 transform/inverse transform of chroma DC coefficients 17

4.3 Scanning and quantization 17
4.3.1 Simple scan 17
4.3.2 Double scan 17
4.3.3 Quantization 18
4.3.4 Scanning and quantization of 2x2 chroma DC coefficients 18

4.4 Use of 2-dimensional model for coefficient coding. 19

4.5 Deblocking filter 19
4.5.1 Block strength 19
4.5.2 Filtering of pixels on both sides of a 4x4 block boundary 20
4.5.3 Filtering across macroblock boundaries connected to intra coding 21

5 ENTROPY CODING 22

5.1 Universal Variable Length Coding (UVLC) 22

5.2 Context-based Adaptive Binary Arithmetic Coding (CABAC) 24
5.2.1 Overview 24
5.2.2 Context Modeling for Coding of Motion and Mode Information 24
5.2.2.1 Context Models for Macroblock Type 25
5.2.2.2 Context Models for Motion Vector Data 26
5.2.2.3 Context Models for Reference Frame Parameter 26
5.2.3 Context Modeling for Coding of Texture Information 27
5.2.3.1 Context Models for Coded Block Pattern 27
5.2.3.2 Context Models for Intra Prediction Mode 27
5.2.3.3 Context Models for Run/Level 27
Block Type 27
5.2.4 Binarization of Non-Binary Valued Symbols 28
Bin_no 28
Bin_no 28
5.2.5 Adaptive Binary Arithmetic Coding 29

6 TEST MODEL ISSUES 30

6.1 Motion Estimation and Mode Decision 30
6.1.1 Low-complexity mode 30
6.1.1.1 Finding optimum prediction mode 30
6.1.1.2 Encoding on macroblock level 31
6.1.2 High-complexity mode 32
6.1.2.1 Motion Estimation 32
6.1.2.2 Mode decision 34
6.1.2.3 Algorithm for motion estimation and mode decision 35

6.2 Quantization 36

6.3 Elimination of single coefficients in inter macroblocks 36
6.3.1 Luma 36
6.3.2 Chroma 37

7 B-PICTURES 38

7.1 Introduction 38

7.2 Five Prediction modes 38

7.3 Finding optimum prediction mode 39

7.4 Syntax 39
7.4.1 Picture type (Ptype) and RUN 40
7.4.2 Macro block type (MB_type) 40
7.4.3 Intra prediction mode (Intra_pred_mode) 41
7.4.4 Reference Frame (Ref_frame) 41
7.4.5 Block Size (Blk_size) 42
7.4.6 Motion vector data (MVDFW, MVDBW) 42

7.5 Decoder Process for motion vector 43
7.5.1 Differential motion vectors 43
7.5.2 Motion vectors in direct mode 43

8 DATA PARTITIONING AND INTERIM FILE FORMAT 43

8.1 Data Partitioning 43

8.2 Interim File Format 44

9 NETWORK ADAPTATION LAYER FOR IP NETWORKS 44

9.1 Assumptions 44

9.2 Combining of Partitions according to Priorities 45

9.3 Packet Structure 45

9.4 Packetization Process 45

9.5 De-packetization 46

9.6 Repair and Error Concealment 46

1 Scope

This document is a description of a reference coding method to be used for the development of a new compression method ITU-T recommendation- H.26L. The basic configuration of the algorithm is similar to H.263.

Some of the differences from H.263 are:

- Only one regular VLC is used for symbol coding
- 1/4 pixel positions are used for motion prediction
- A number of different blocksizes are used for motion prediction
- Residual coding is based on 4x4 blocks and a integer transform is used
- Multiple reference frames may be used for prediction and this is considered to replace any use of B-frames

The first part of the document describes the coding method by mainly defining the decoder actions. Towards the end, 'test model issues' relevant for the encoder to be used as reference for the development of the standard will be covered. This split should make it easy at a later stage to split the document into what is relevant for the standard and a test model. However, at the moment we find it preferable to have one combined document.

2 Organization of source- and compressed data

2.1 Picture formats

At the moment only the QCIF and CIF formats are included in the model [Not true of the software anymore – the document needs to be updated about this (but not relevant to testing at QCIF and CIF resolutions)].

2.2 Subdivision of a picture into macroblocks

A CIF picture is divided into 18x22 = 396 macroblocks. Similarly a QCIF picture is divided into 99 macroblocks as indicated in Figure 1. At the moment there are no other layers in the described model. [Slice layers are supported in the software, but not fully reflected in this document.]

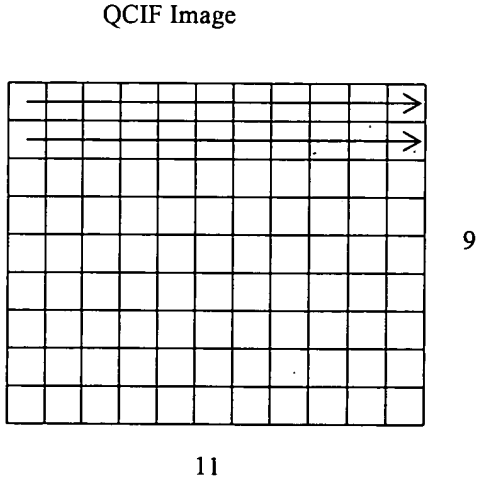


Figure 1 Subdivision of a QCIF picture into 16x16 macroblocks

2.3 *Order of the bitstream within a macroblock*

Figure 2 and Figure 3 indicates how a macroblock is divided and the order of the different syntax elements resulting from coding a macroblock.

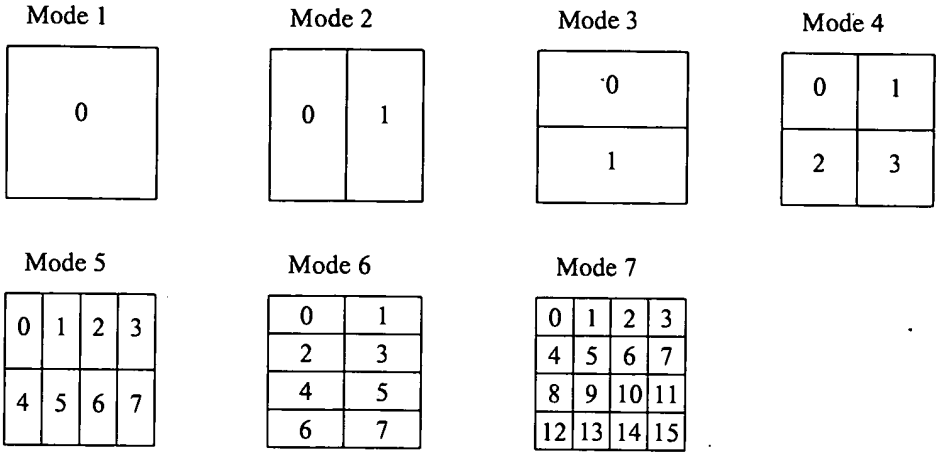


Figure 2 Numbering of the vectors for the different blocks depending on the inter mode. For each block the horizontal component comes first followed by the vertical component

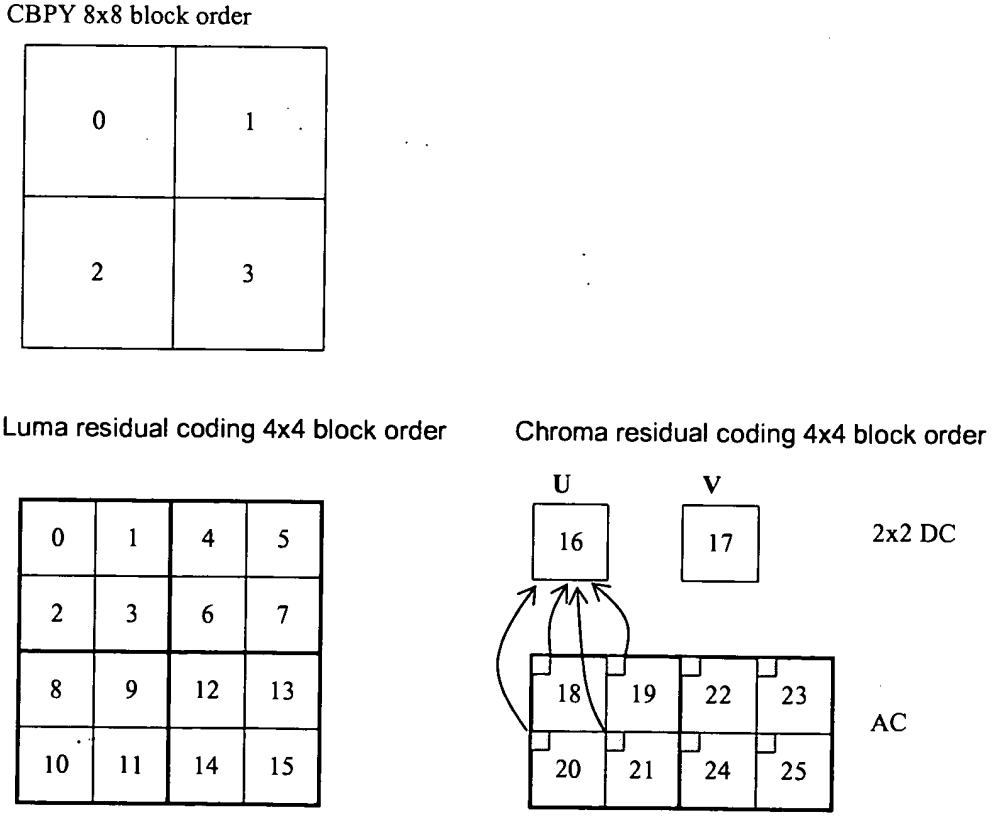


Figure 3 Ordering of blocks for CBPY and residual coding of 4x4 blocks

2.4 *Syntax*

2.4.1 Syntax diagram

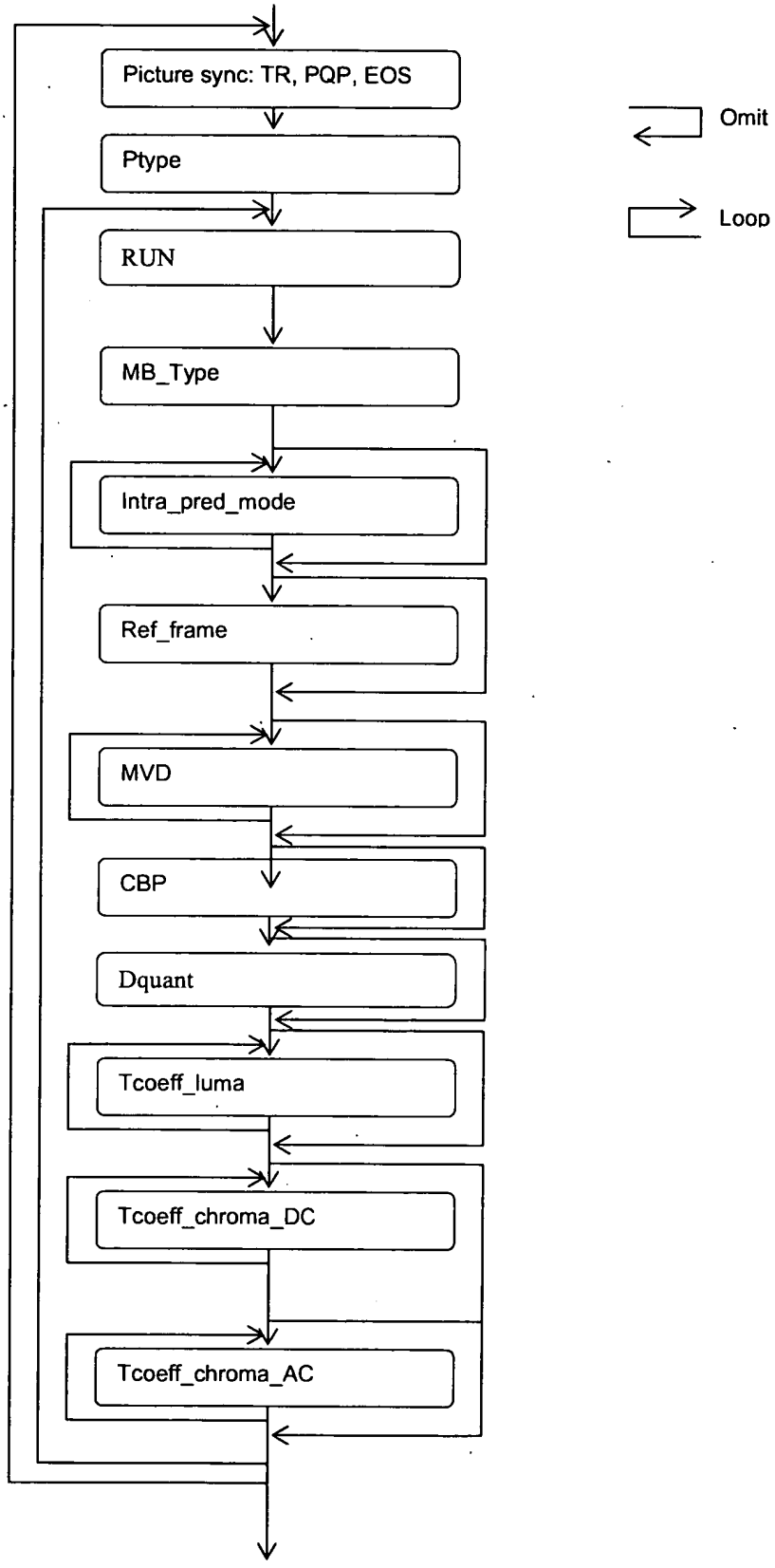


Figure 4 Syntax diagram for all the elements in the video bitstream

3 Description of syntax elements

3.1 Picture sync

The first codeword in a picture is 31 bits long (L = 31). It works as a picture sync, but it also contains an INFO part that has 15 bits. These bits are used for:

TR(8 bits) Temporal reference. The value of TR is formed by incrementing its value in the temporally-previous reference picture header by one plus the number of skipped or non-reference pictures at the picture clock frequency since the previously transmitted one.

PQP(5 bits) Picture QP: Information about the quantizer QUANT to be used for luma for the picture. (See under Quantization concerning QUANT for chroma). The 5 bit representation is the natural binary representations of the values of QP which range from 0 to 31. QP is a pointer to the actual quantization parameter QUANT to be used. (See below under quantization). The range of quantization value is still about the same as for H.263, 1-31. An approximate relation between the QUANT in H.263 and QP is: $QUANT_{H.263}(QP) \approx QP_0(QP) = 2^{QP/6}$. $QP_0()$ will be used later for scaling purposes when selecting prediction modes.

Formats 0 indicates QCIF, 1 indicates CIF

EOS 0 for picture header. 1 indicates End Of Sequence

The sketch below indicates where the different bits are located within INFO.

8b: TR	5b: PQP	1b: Formats	1b: EOS

3.2 Picture type (Ptype)

Code_number =0: Inter picture with prediction from the most recent decoded picture only.

Code_number =1: Inter picture with possibility of prediction from more than one previous decoded picture. For this mode information reference picture for prediction must be signalled for each macroblock.

Code_number =2: Intra picture.

Code_number =3: B picture with prediction from the most recent previous decoded and subsequent decoded pictures only.

Code_number =4: B picture with possibility of prediction from more than one previous decoded picture and subsequent decoded picture. When using this mode, information reference frame for prediction must be signaled for each macroblock.

3.3 RUN

A macroblock is called skipped if no information is sent. In that case the reconstruction of a n inter macroblock is made by copying the colocated picture material from the laset decoded frame. (See separate definition for B-pictures). RUN indicates the number of skipped macroblocks in an inter- or B-picture before a coded picture. If the last MB of the frame is not coded an additional codeword is added which points to a MB outside the picture.

3.4 Macro block type (MB_Type)

Refer to **Error! Reference source not found.**. There are different MB-Type tables for Intra and Inter frames.

3.4.1 Intra

Intra 4x4 Intra coding as defined in sections 3.5.1 to 3.5.8.

Imode, nc, AC	See definition in section 3.5.9.3. These modes refer to 16x16 intra coding.

3.4.2 Inter

Skip	No further information about the macroblock is transmitted. A copy of the colocated macroblock in the most recent decoded picture is used as reconstruction for the present macroblock.
NxM (eg. 8x4)	The macroblock is predicted from a past picture with block size NxM. For each NxM block motion vector data is provided. Depending on N and M there may be 1 to 16 sets of motion vector data for a macroblock.
Intra 4x4	4x4 intra coding.

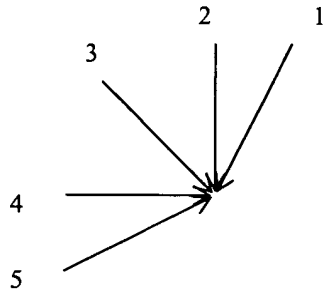
Code numbers from 9 and upwards represent 16x16 intra coding.

3.5 Intra prediction mode (Intra_pred_mode)

Even in Intra mode, prediction is always used for each sub block in a macroblock. A 4x4 block is to be coded (pixels labelled a to p below). The pixels A to I from neighbouring blocks are already decoded and may be used for prediction.

I	A	B	C	D
E	a	b	c	d
F	e	f	g	h
G	i	j	k	l
H	m	n	o	p

There are 6 intra prediction modes labelled 0 to 5. Mode 0 is 'DC-prediction' (see below). The other modes represent directions of predictions as indicated below.



3.5.1 Mode 0: DC prediction

Generally all pixels are predicted by (A+B+C+D+E+F+G+H)//8. If four of the pixels are outside the picture, the average of the remaining four is used for prediction. If all 8 pixels are outside the picture the prediction for all pixels in the block is 128. A block may therefore always be predicted in this mode.

3.5.2 Mode 1:

This mode is used only if all A,B,C,D are inside the picture.

a is predicted by:	(A+B)/2
e is predicted by	B
b,i are predicted by	(B+C)/2
f,m are predicted by	C
c,j are predicted by	(C+D)/2
d,g,h,k,l,n,o,p are predicted by	D

3.5.3 Mode 2: Vertical prediction

If A,B,C,D are inside the picture, a,e,i,m are predicted by A, b,f,j,n by B etc.

3.5.4 Mode 3: Diagonal prediction

This mode is used only if all A,B,C,D,E,F,G,H,I are inside the picture. This is a 'diagonal' prediction.

m is predicted by:	(H+2G+F)//4
i,n are predicted by	(G+2F+E)//4
e,j,o are predicted by	(F+2E+I)//4
a,f,k,p are predicted by	(E+2I+A)//4
b,g,l are predicted by	(I+2A+B)//4
c,h are predicted by	(A+2B+C)//4
d is predicted by	(B+2C+D)//4

3.5.5 Mode 4:Horizontal prediction

If E,F,G,H are inside the picture, a,b,c,d are predicted by E, e,f,g,h by F etc.

3.5.6 Mode 5:

This mode is used only if all E,F,G,H are inside the picture.

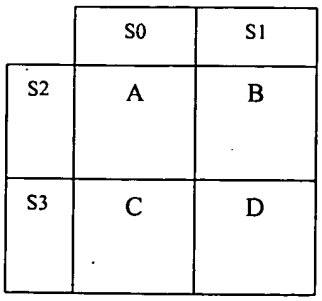
a is predicted by:	(E+F)/2
b is predicted by	F
c,e are predicted by	(F+G)/2
f,d are predicted by	G
i,g are predicted by	(G+H)/2
h,j,k,l,m,n,o,p are predicted by	H

3.5.7 Prediction of chroma blocks

For chroma prediction there is only one mode. No information is therefore needed to be transmitted. The prediction is indicated in the figure below. The 8x8 chroma block consists of 4 4x4 blocks A,B,C,D. S0,1,2,3 are the sums of 4 neighbouring pixels.

If S0, S1, S2, S3 are all inside the frame:
A = (S0 + S2 + 4)/8
B = (S1 + 2)/4
C = (S3 + 2)/4
D = (S1 + S3 + 4)/8

If only S0 and S1 are inside the frame:
A = (S0 + 2)/4
B = (S1 + 2)/4
C = (S0 + 2)/4
D = (S1 + 2)/4



If only S2 and S3 are inside the frame:
A = (S2 + 2)/4
B = (S2 + 2)/4
C = (S3 + 2)/4
D = (S3 + 2)/4
If S0, S1, S2, S3 are all outside the frame: A = B = C = D = 128
(Note: This prediction should be considered changed)

3.5.8 Coding of Intra prediction modes

Since each of the 4x4 luma blocks shall be assigned a prediction mode, this will require a considerable number of bits if coded directly. We have therefore tried to find more efficient ways of coding mode information. First of all we observe that the chosen prediction of a block is highly correlated with the prediction modes of adjacent blocks. This is illustrated in Figure 5 a. When the prediction modes of A and B are known (including the case that A or B or both are outside the picture) an ordering of the most probable, next most probable etc. of C is given. This ordering is listed in Table 1. For each prediction mode of A and B a list of 5 numbers is given. Example: Prediction mode for A and B is 2. The string 2 1 0 3 4 5 indicates that mode 2 is also the most probable mode for block C. Mode 1 is the next most probable one etc. In the bitstream there will for instance be information that Prob0 = 1 (see **Error! Reference source not found.**) indicating that the next most probable mode shall be used for block C. In our example this means Intra prediction mode 1. Use of '–' in the table indicates that this instance can not occur because A or B or both are outside the picture.
For more efficient coding, information on intra prediction of two 4x4 luma blocks are coded in one codeword (Prob0 and Prob1 in **Error! Reference source not found.**). The order of the resulting 8 codewords is indicated in Figure 5 b.

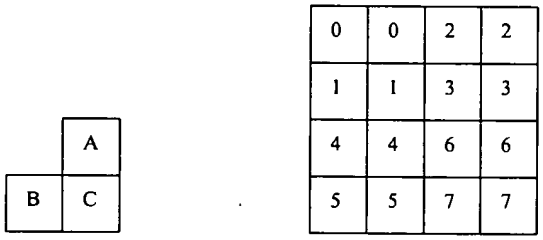


Figure 5 a) Prediction mode of block C shall be established. A and B are adjacent blocks. b) order of intra prediction information in the bitstream

Table 1 Prediction mode as a function of ordering signalled in the bitstream (see text)

B\A	outside	0	1	2	3	4	5
outside	0 -----	021 ---	102 ---	201 ---	012 ---	012 ---	012 ---
0	045 ---	041352	104325	230415	304215	043152	043512
1	045 ---	014325	102435	203145	032145	041325	014352
2	045 ---	012345	102345	210345	302145	042135	013245
3	045 ---	304152	310425	231054	304215	403512	305412
4	405 ---	403512	401532	240351	430512	403512	405312
5	504 ---	540312	015432	201453	530412	450312	504132

3.5.9 Intra mode based on 16x16 macroblocks (16x16 intra mode)

This intra mode is particularly suitable for regions with little details, also referred to as 'flat' regions.

3.5.9.1 Prediction modes

Assume that the block to be predicted has pixel locations 0 to 15 horizontally and 0 to 15 vertically. We use the notation P(i,j) where i,j = 0..15. P(i,-1), i=0..15 are the neighboring pixels above the block and P(-1,j), j=0..15 are the neighboring pixels to the left of the block. Pred(i,j) i,j = 0..15 is the prediction for the whole luma macroblock. We have 4 different prediction modes:

IMODE = 0 (vertical)

Pred(i,j) = P(i,-1), i,j=0..15

IMODE = 1 (horizontal)

Pred(i,j) = P(-1,j), i,j=0..15

IMODE = 2 (DC prediction)

$$\text{Pred(i,j)} = (\sum_{i=0}^{15}(P(-1,i)+P(i,-1)))/32 \quad \text{i,j=0..15}$$

IMODE = 3 (Plane prediction)

Pred(i,j) = (a + bx(i-7) + cx(j-7) +16)/32

Where:

a = 16x(P(-1,15) + P(15,-1))

b = 5x(*H*/4)/16

c = 5x(*V*/4)/16

$$H = \sum_{i=1}^{8} ix(P(7+i,-1)+P(7-i,-1))$$

$$V = \sum_{j=1}^{8} jx(P(-1,7+j)+P(-1,7-j))$$

3.5.9.2 Residual coding

The residual coding is based on 4x4 transform. But similar to coding of chroma coefficients, another 4x4 transform to the 16 DC coefficients in the macroblock are added. In that way we end up with an overall DC for the whole MB which works well in flat areas.

Since we use the same integer transform to DC coefficients, we have to perform additional normalization to those coefficients, which implies a division by 676. To avoid the division we performed normalization by $49/2^{15}$ on the encoder side and $48/2^{15}$ on the decoder side, which gives sufficient accuracy.

Only single scan is used for 16x16 intra coding.

To produce the bitstream, we first scan through the 16 'DC transform' coefficients. There is no 'CBP' information to indicate no coefficients on this level. If AC = 1 (see below) ac coefficients of the 16 4x4 blocks are scanned. There are 15 coefficients in each block since the DC coefficients are included in the level above.

3.5.9.3 Signalling of mode information for 16x16 intra coding

See **Error! Reference source not found.**. Three parameters have to be signaled. They are all included in MB-type.

Imode: 0,1,2,3

AC: 0 means there are no ac coefficients in the 16x16 block. 1 means that there is at least one ac coefficient and all 16 blocks are scanned.

nc: CBP for chroma (see 3.8)

3.6 Reference frame (Ref_frame)

If PTYPE indicates possibility of prediction from more than one previous decoded picture, the exact frame to be used must be signalled. This is done according to the following table.

Code_number	Reference frame
0	The last decoded frame (1 frame back)
1	2 frames back
2	3 frames back
..	..

3.7 *Motion Vector Data (MVD)*

If so indicated by MB_type, vector data for 1-16 blocks are transmitted. For every block a prediction is formed for the horizontal and vertical components of the motion vector. MVD signals the difference between the vector component to be used and this prediction. The order in which vector data is sent is indicated in Figure 2. Motion vectors are allowed to point to pixels outside the reference frame. If a pixel outside the reference frame is referred to in the prediction process the nearest pixel belonging to the frame (an edge or corner pixel) shall be used. This process is performed on a pixel bases..

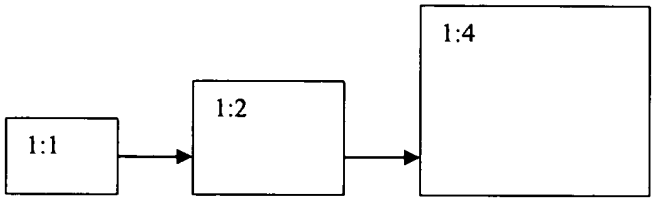
3.7.1 Fractional pixel accuracy

3.7.1.1 1/4 pixel accuracy

Note: It has been adopted to use direct interpolation to produce 1/4 pixel values (see VCEG-L20) at the decoder. However, this also has implications on the encoder which must use extra effort to avoid encoder/decoder mismatch. This issue will be further clarified in an ad.hoc. group and a solution presented at the next meeting. The text will then be updated.

In the baseline coding mode, 1/4 pixel accuracy is used to define motion vectors. To avoid encoder/decoder mismatch, the interpolation procedure has to be defined in detail.

The figure below indicates that interpolation for luminance may be defined as taking place in two steps.



Interpolation: 6H, 6V bilinear

Step I: Generation of ½ pixel positions

The 'x' - es below represent integer pixel positions. First the '+' - es (horizontal ½ pixel positions) are produced by using a 6 tap filter: (1,-5,20,20,-5,1)/32. The result is rounded to the nearest integer and clipped to the range 0 to 255. Then all the '*' – es are produced by using the same 6 tap. The result is rounded to the nearest integer and clipped to the range 0 to 255. If the filtering process refer to a pixel outside the picture, it is replaced by the nearest picture edge pixel.

```
x + x + x + x + x
* * * * * * * * *
x + x + x + x + x
* * * * * * * * *
x + x + x + x + x
* * * * * * * * *
```

Step II: Generation of ¼ pixel positions

The 'x' - es below represent ½ pixel positions (from above). First the '+' - es are produced by linear interpolation (average of the neighbouring ½ pixel positions). The result is truncated to the nearest integer. Then all the '*' – es are produced by vertical linear interpolation with truncation to the nearest integer.

```
x + x + x + x + x
* * * * * * * * *
x + x + x + x + x
* * * * * * * * *
x + x + x + x + x
* * * * * * * * *
```

Interpolation position with more low pass filtering

The figure below indicate a grid with interpolated positions where capital letters represent integer positions, numbers represent ½ pixel positions and lower case letters represent 1/4 pixel positions.

```
A a 1 b B
c d e f g
2 h 3 i 4
j k l m n
C o 5 p D
```

Relative to integer positions, the position m is chosen to represent a position with more low pass filtering. According to the definition above, m ≈ ('3' + '4' + '5' + 'D')/4. (≈ is used due to possible rounding effects). Instead we define:

m = ('A' + 'B' + 'C' + 'D' + 2)/4

3.7.1.2 1/8 pixel accuracy

For a higher complexity or higher coding efficiency profile (not yet defined) 1/8 pixel accuracy prediction using 8-tap filters may be used. The interpolation process is assumed to be performed in a separable manner - first in the horizontal direction and then in the vertical direction. Hence the filter operation is fully specified by the one-dimensional filter taps. The filter taps are given below:

Position								
Integer				1				
1/8	(-3	12	-37	485	71	-21	6	-1)/512
2/8	(-3	12	-37	229	71	-21	6	-1)/256
3/8	(-6	24	-76	387	229	-60	18	-4)/512
4/8	(-3	12	-39	158	158	-39	12	-3)/256
5/8	(-4	18	-60	229	387	-76	24	-6)/512
6/8	(-1	6	-21	71	229	-37	12	-3)/256
7/8	(-1	6	-21	71	485	-37	12	-3)/512

3.7.2 Prediction of vector components

In all macroblocks with square motion vector blocks (16x16, 8x8, 4x4) "median prediction" (see 3.7.2.1) is used. In case the macroblock may be classified to have directional segmentation the prediction is defined in 3.7.2.2.

3.7.2.1 Median prediction

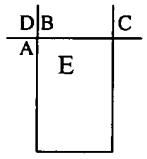
In the figure below the vector component E of the indicated block shall be predicted. The prediction is normally formed as the median of A, B and C. However, the prediction may be modified as described below. Notice that it is still referred to as "median prediction"

A The component applying to the pixel to the left of the upper left pixel in E

B The component applying to the pixel just above the upper left pixel in E

C The component applying to the pixel above and to the right of the upper right pixel in E

D The component applying to the pixel above and to the left of the upper left pixel in E



A, B, C, D and E may represent motion vectors from different reference pictures. As an example we may be seeking prediction for a motion vector for E from the last decoded picture. A, B, C and D may represent vectors from 2, 3, 4 and 5 pictures back. The following substitutions may be made prior to median filtering.

- If A and D are outside the picture, their values are assumed to be zero and they are considered to have "different reference picture than E".
- If D, B, C are outside the picture, the prediction is equal to A (equivalent to replacing B and C with A before median filtering).

- If C is outside the picture or still not available due to the order of vector data (see Figure 2), C is replaced by D.

If any of the blocks A, B, C, D are intra coded they count as having "different reference frame. If one and only one of the vector components used in the median calculation (A, B, C) refer to the same reference picture as the vector component E, this one vector component is used to predict E.

3.7.2.2 Directional segmentation prediction

If the macroblock where the block to be predicted belongs to has a directional segmentation (vector block size of 8x16, 16x8, 8x4 or 4x8) the prediction is generated as follows (refer to figure below and the definitions of A, B, C, E above):

Vector block size 8x16:

Left block:	A is used as prediction if it has the same reference picture as E, otherwise "Median prediction" is used
Right block:	C is used as prediction if it has the same reference picture as E, otherwise "Median prediction" is used

Vector block size 16x8:

Upper block:	B is used as prediction if it has the same reference picture as E, otherwise "Median prediction" is used
Lower block:	A is used as prediction if it has the same reference picture as E, otherwise "Median prediction" is used

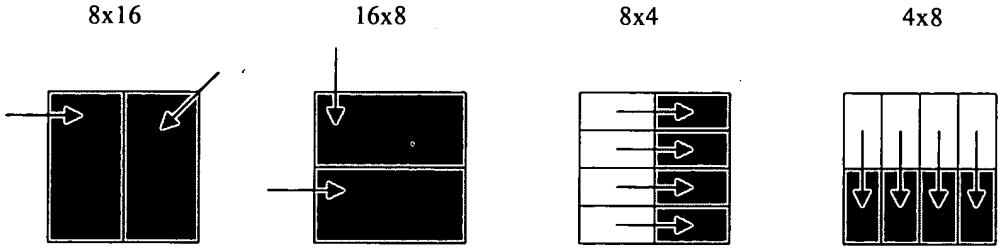
Vector block size 8x4:

For white blocks:	"Median prediction" is used
For shaded blocks:	A is used as prediction

Vector block size 4x8:

For white blocks:	"Median prediction" is used
For shaded blocks:	B is used as prediction

If the indicated prediction block is outside the picture, the same substitution rules are applied as in the case of median prediction.



3.7.3 Chroma vectors

Chroma vectors are derived from the luma vectors. Since chroma has half resolution compared to luma, the chroma vectors are obtained by a division of two:

Croma_vector = Luma_vector/2 - which means that the chroma vectors have a resolution of 1/8 pixel.

Due to the half resolution, a chroma vector applies to 1/4 as many pixels as the luma vector. For example if the luma vector applies to 8x16 luma pixels, the corresponding chroma vector applies to 4x8 chroma pixels and if the luma vector applies to 4x4 luma pixels, the corresponding chroma vector applies to 2x2 chroma pixel.

For fractional pixel interpolation for chroma prediction, bilinear interpolation is used. The result is rounded to the nearest integer.

3.8 Coded Block Pattern (CBP)

The CBP contains information of which 8x8 blocks - luma and chroma - contain transform coefficients. Notice that an 8x8 block contains 4 4x4 blocks meaning that the statement '8x8 block contains coefficients' means that 'one or more of the 4 4x4 blocks contain coefficients'. The 4 least significant bits of CBP contain information on which of 4 8x8 luma blocks in a macroblock contains nonzero coefficients. Let us call these 4 bits CBPY. The ordering of 8x8 blocks is indicated in Figure 3. A 0 in position n of CBP (binary representation) means that the corresponding 8x8 block has no coefficients whereas a 1 means that the 8x8 block has one or more non-zero coefficients.

For chroma we define 3 possibilities:

nc=0: no chroma coefficients at all.

nc=1 There are nonzero 2x2 transform coefficients. All chroma AC coefficients = 0. Therefore we do not send any EOB for chroma AC coefficients.

nc=2 There may be 2x2 nonzero coefficients and there is at least one nonzero chroma AC coefficient present. In this case we need to send 10 EOBs (2 for DC coefficients and 2x4=8 for the 8 4x4 blocks) for chroma in a macroblock.

The total CBP for a macroblock is: **CBP = CBPY + 16xnc**

The CBP is signalled with a different codeword for Inter macroblocks and Intra macroblocks since the statistics of CBP values are different in the two cases.

3.9 Dquant

Dquant contains the possibility of changing QUANT on the macroblock level. It does not need to be present for macroblocks without nonzero transform coefficients. More specifically Dquant is present for non-skipped macroblocks:

- If CBP indicates that there are nonzero transform coefficients in the MB

or

- If the MB is 16x16 based intra coded

Tha value of Dquant shall be interpreted in the same way as Motion Vecor Data (**Error! Reference source not found.**). Its value may range from -16 to +16 which enables the QP to be changed to any value in the range 0-31.

$QUANT_{new} = modulo_{32}(QUANT_{old} + Dquant + 32)$ (also known as "arithmetic wrap")

4 Transform and inverse transform

This section defines all elements related to transform coding and decoding. It is therefore relevant to all the syntax elements 'Tcoeff' in the syntax diagram.

4.1 4x4 block size

Instead of DCT, an integer transform with basically the same coding property as a 4x4 DCT is used. The transformation of the pixels a,b,c,d into 4 transform coefficients A,B,C,D is defined by:

```
A =  13a + 13b + 13c + 13d
B =  17a +  7b -  7c - 17d
C =  13a - 13b - 13c + 13d
D =   7a - 17b + 17c -  7d
```

The inverse transformation of transform coefficients A,B,C,D into 4 pixels a',b',c',d' is defined by:

```
a' = 13A + 17B + 13C +  7D
b' = 13A +  7B - 13C - 17D
```

```
c' = 13A -  7B - 13C + 17D
d' = 13A - 17B + 13C -  7D
```

The relation between a and a' is: a' = 676a. This is because the expressions defined above contain no normalisation. Normalisation will be performed in the quantization/dequantisation process and a final shift after inverse quantization.

The transform/inverse is performed both vertically and horizontally in the same manner as in H.263. By the above exact definition of inverse transform, the same operations will be performed on coder and decoder side which means that we have no 'inverse transform mismatch'.

4.2 2x2 transform/inverse transform of chroma DC coefficients

With the low resolution of chroma it seems to be preferable to have larger blocksize than 4x4. Specifically the 8x8 DC coefficient seems very useful for better definition of low resolution chroma. The 2 dimensional 2x2 transform procedure is illustrated below. DC0,1,2,3 are the DC coefficients of 2x2 chroma blocks.

DC0	DC1	**Two dimensional 2x2 transform** ⇒	DDC(0,0)	DDC(1,0)
DC2	DC3		DDC(0,1)	DDC(1,1)

Definition of transform:

DCC(0,0) = (DC0+DC1+DC2+DC3)/2

DCC(1,0) = (DC0-DC1+DC2-DC3)/2

DCC(0,1) = (DC0+DC1-DC2-DC3)/2

DCC(1,1) = (DC0-DC1-DC2+DC3)/2

Definition of inverse transform:

DC0 = (DCC(0,0)+ DCC(1,0)+ DCC(0,1)+ DCC(1,1))/2

DC1 = (DCC(0,0)- DCC(1,0)+ DCC(0,1)- DCC(1,1))/2

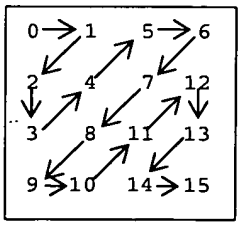
DC2 = (DCC(0,0)+ DCC(1,0)- DCC(0,1)- DCC(1,1))/2

DC3 = (DCC(0,0)- DCC(1,0)- DCC(0,1)+ DCC(1,1))/2

4.3 Scanning and quantization

4.3.1 Simple scan

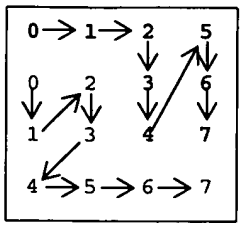
Except for Intra coding of luma with QP<24, simple scan is used. This is basically zig-zag scanning similar to the one used in H.263. The scanning pattern is:



4.3.2 Double scan

When using the VLC defined above, we use a one bit code for EOB. For Inter blocks and Intra with high QP the probability of EOB is typically 50% which is well matched with the VLC. In other words this means

that we on average have one non-zero coefficient per 4x4 block in addition to the EOB code (remember that a lot of 4x4 blocks only have EOB). On the other hand, for Intra coding we typically have more than one non-zero coefficient per 4x4 block. This means that the 1 bit EOB becomes inefficient. To improve on this the 4x4 block is subdivided into two parts that are scanned separately and with one EOB each. The two scanning parts are shown below – one of them in bold.



4.3.3 Quantization

The quantization/dequantization process shall perform 'normal' quantization/dequantization as well as take care of the fact that transform operations are kept very simple and therefore do not contain normalization of transform coefficients. 32 different QP values are used. They are arranged so that there is an increase of step size of about 12% from one QP to the next. Increase of QP by 6 means that the step size is about doubled. There is no 'dead zone' in the quantization process and the total range of step size from smallest to largest is about the same as for H.263.

The QP signalled in the bitstream applies for luma quantization/dequantization. This could be called QP_{luma}. For chroma quantization/dequantization a different value - QP_{chroma} - is used. The relation between the two is:

QP_{luma}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
QP_{chroma}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	17	18	19	20	20	21	22	22	23	23	24	24	25	25

When QP is used in the following we mean QP_{luma} or QP_{chroma} depending on what is appropriate.

Two arrays of numbers are used for quantization/dequantization.

A(QP=0,..,31)
620,553,492,439,391,348,310,276,246,219,195,174,155,138,123,110,98,87,78,69,62,55,49,44,39,35,
31,27,24,22,19,17

B(QP=0,..,31)

3881,4351,4890,5481,6154,6914,7761,8718,9781,10987,12339,13828,15523,17435,19561,21873,24552,27656,308
47,34870,38807,43747,49103,54683,61694,68745,77615,89113,100253,109366,126635,141533

The relation between A() and B() is: $A(QP) \times B(QP) \times 676^2 = 2^{40}$.

It is assumed that a coefficient K is quantized in the following way:

$LEVEL = (KxA(QP) + fx2^{20})/2^{20}$ $|f|$ is in the range (0-0.5) and f has the same sign as K.

Dequantization:

K' = LEVELxB(QP)

After inverse transform this results in pixel values that are 2^{20} too high. A shift of 20 bits (with rounding) is therefore needed on the reconstruction side. The definition of transform and quantization is designed so that no overflow will occur with use of 32 bit arithmetic.

4.3.4 Scanning and quantization of 2x2 chroma DC coefficients

DDC() is quantized (and dequantized) separately resulting in LEVEL, RUN and EOB. The scanning order is: DCC(0,0), DCC(1,0), DCC(0,1), DCC(1,1). Inverse 2x2 transform as defined above is then performed after dequantization resulting in dequantized 4x4 DC coefficients: DC0' DC1' DC2' DC3'.

Chroma AC coefficients (4x4 based) are then quantized similarly to before. Notice that there are only 15 AC coefficients. The maximum size of RUN is therefore 14. However, for simplicity we use the same relation between LEVEL, RUN and Code no. As defined for 'Simple scan' in **Error! Reference source not found.**.

4.4 Use of 2-dimensional model for coefficient coding.

In the 3D model for coefficient coding (see H.263) there is no good use of a short codeword of 1 bit. On the other hand, with the use of 2D VLC plus End Of Block (EOB) (as used in H.261, H.262) and with the small block size, 1 bit for EOB is usually well matched to the VLC.

Furthermore, with the fewer non-zero coefficients per block, the advantage of using 3D VLC is reduced.

As a result we use a 2D model plus End Of Block (EOB) in the present model. This means that an event to be coded (RUN, LEVEL) consists of:

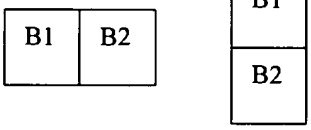
RUN which is the number of zero coefficients since the last nonzero coefficient.

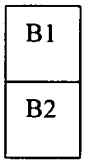
LEVEL the size of the nonzero coefficient

EOB signals that there are no more nonzero coefficients in the block

4.5 Deblocking filter

The deblocking filter process takes place both across vertical block edges and horizontal block edges. The filtering is performed first across vertical edges. The filtered pixels are then used when performing filtering across horizontal block edges. As a result block1 and block2 (see below) may be located in one of the positions:

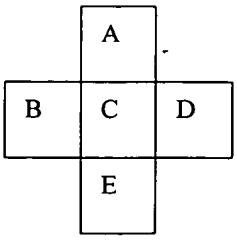




The filtering may be based on 4x4 block boundaries or macroblock boundaries if one or both adjacent macroblocks are intra coded (see 4.5.3)

4.5.1 Block strength

Here we consider filteringbased on a 4x4 block structure. Each 4x4 block in a reconstructed picture is assigned a **Strength** value ranging from 0 to 3. This value is assigned based on different criteria. The maximum value is chosen. Assume that one criterion results in **value** to be assigned. The new assignment is: **Strength = max(Strength,value),** so that we always keep the maximum value. In the figure below A, B, C, D and E are 4x4 blocks. Blocks may be luma or chroma. Different strenght is used for luma and chroma. This difference is introduced by $\mathbf{St_0}$ which is 0 for chroma and 1 for luma.



The procedure for assignment is as follows:

- Initially each 4x4 block of a picture is assigned **Strength = 0**
- If C is intra coded: $\mathbf{Strength_C = St_0 + 2}$, $\mathbf{Strength_A = max(Strength_A, St_0 + 1)}$, $\mathbf{Strength_B = max(Strength_B, St_0 + 1)}$, $\mathbf{Strength_D = max(Strength_D, St_0 + 1)}$, $\mathbf{Strength_E = max(Strength_E, St_0 + 1)}$
- If C has nonzero transform coefficients: $\mathbf{Strength_C = max(Strength_C,2)}$, $\mathbf{Strength_A = max(Strength_A,1)}$, $\mathbf{Strength_B = max(Strength_B,1)}$, $\mathbf{Strength_D = max(Strength_D,1)}$, $\mathbf{Strength_E = max(Strength_E,1)}$

- For luma blocks: If the absolute difference between one of the vector components of block A and block C is at least one integer pixel (four ¼ pixels) then: **$Strength_A$ = max($Strength_A$,1), $Strength_C$ = max($Strength_C$,1)**. The same procedure is applied to vector differences between blocks (B and C), (C and D) and (C and E)
- For chroma blocks: Since chroma has half the resolution of luma, there may be different vectors also inside the chroma block C. This is illustrated by the horizontal broken line i block C. The procedure for a chroma block is that if the absolute difference between one of the vector components on each side of the boundary between A and C or across the dotted line inside C is at least one integer pixel (four ¼ pixels) then: **$Strength_A$ = max($Strength_A$,1), $Strength_C$ = max($Strength_C$,1)**. A similar procedure is applied to vector differences between blocks (B and C), (C and D) and (C and E) *(Note: This description for chroma may seem strange. It is written to reflect what happens in the software. A simpler rule should be sought and reflected in a later version of TML as well as in the software)*

4.5.2 Filtering of pixels on both sides of a 4x4 block boundary

In the following description it is assumed that a vertical 4x4 grid boundary is to be filtered (that is, horizontal filtering takes place). Horizontal grid boundaries are filtered utilizing the same algorithm, but 'left' pixels should be substituted for 'above' pixels and 'right' pixels for 'below' pixels. Let us denote the set of eight pixels across a 4x4 block boundary as l_4, l_3, l_2, l_1, r_1, r_2, r_3, r_4.
Up to two pixels can be updated as a result of the filtering process on both sides of the boundary (that is at most l_2, l_1, r_1, r_2). The number of updated pixels on both sides of the boundary depends on corresponding right and left activity parameters a_r and a_l. Both activity parameters are limited by the so called overall activity parameter *n*:

$$n = \begin{cases} 1 & \alpha + \alpha/2 \le 2\Delta \\ 2 & \alpha \le 2\Delta < \alpha + \alpha/2 \\ 3 & 2\Delta < \alpha \end{cases}$$

where $\Delta = |r_1 - l_1|$ is a positive integer representing gray level difference across the boundary, and α is a QP dependent threshold specified in Table 1.

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
α	0	0	0	0	0	0	0	0	3	3	3	7	7	7	12	17	17	22	28	34	40	47	60	67	82	89	112	137	153	187	213	249
β	0	0	0	0	0	0	0	0	3	3	3	4	4	4	6	6	6	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14

Table 1. QP dependent threshold parameters α and β.

The right filter activity parameter a_r is equal to the smallest integer i>0 not larger than *n* for which the inequality $|r_1 - r_{i+1}| * i > \beta$ holds. The QP dependent threshold β is given in Table 1. If no $i < n$ satisfies the condition, a_r is set to *n*. The left filter activity parameter a_l is obtained using the same set of rules substituting pixels on the right of the boundary r_i with pixels l_i on the left of the boundary. Once a_r and a_l are calculated, the pixels to be updated and their filtering support windows are determined by using Table 2. The same table applies for both right and left pixel although only right pixels are listed in the table. Information on how to filter the left pixels is obtained by swapping a_r and a_l and substituting r_1, r_2 with l_1, l_2.

a_r (only if a_l>1)	pixel r_1	pixel r_2
1	no filtering	no filtering
2	filtered	no filtering
3	filtered	filtered

Table 2. Pixels to be updated as a function of a_r.

When filtering within P frames, the left (right) activity parameter is upper bounded by 2 if *strength* of the left (right) 4x4 block is no larger than 1. Selected pixels are filtered as follows:

$$L_1 = (21*l_2 + 22*l_1 + 21*r_1 + 32)/64$$
$$L_2 = (21*l_3 + 22*l_2 + 21*L_1 + 32)/64$$
$$R_1 = (21*l_1 + 22*r_1 + 21*r_2 + 32)/64$$
$$R_2 = (21*R_1 + 22*r_2 + 21*r_3 + 32)/64$$

When either of pixels r_1, l_1 are updated by filtering then the difference between filtered and nonfiltered value is clipped to the range [-C, C] where C = (Clip(QP,strength$_{left}$) + Clip(QP,strength$_{right}$) + a_r + a_l) / 2. When pixel r_2 is updated by filtering then the difference is clipped to the range [-C_{right}, C_{right}] with C_{right} = Clip(QP,strength$_{right}$). Similarly for pixel l_2.

Definition of the table Clip(QP,strength):

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	29	30	31
Clip(qp,1)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3
Clip(qp,2)	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4	5	5
Clip(qp,3)	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	4	4	5	5	5	7	8	9

The filter is applied on a 4x4 grid for both luminance and chrominace, first for all the vertical then for all the horizontal block boundaries.

4.5.3 Filtering across macroblock boundaries connected to intra coding

In case of intra coding and areas in the picture with little texture the subjective quality may be improved by stronger filtering across macroblock boundaries. This stronger filtering across a macroblock boundary is performed if:

One or both of the adjacent macroblocks are intra coded

and

a_r and a_l are equal to 3

and

Δ is in the range [2-qp/4]

In case these conditions are fulfilled, the pixels l_3 to r_3 when filtering luminance frames and l_2 to r_2 when filtering chrominance frames are modified in the following way:

$$L_1 = (25*l_3 + 26*l_2 + 26*l_1 + 26*r_1 + 25*r_2 + 64)/128$$
$$L_2 = (25*l_4 + 26*l_3 + 26*l_2 + 26*L_1 + 25*r_1 + 64)/128$$
$$L_3 = (\quad 26*l_4 + 51*l_3 + 26*L_2 + 25*L_1 + 64)/128$$
$$R_1 = (25*l_2 + 26*l_1 + 26*r_1 + 26*r_2 + 25*r_3 + 64)/128$$
$$R_2 = (25*l_1 + 26*R_1 + 26*r_2 + 26*r_3 + 25*r_4 + 64)/128$$
$$R_3 = (25*R_1 + 26*R_2 + 51*r_3 + 26*r_4 \quad + 64)/128$$

Notice that when filtering across the next 4x4 block boundary, pixel r_3 may get overwritten. To compensate the effect R_3 is stored and used as l_2 when filtering across the next 4x4 block boundary.

Otherwise filtering is done as described in the section above.

5 Entropy Coding

5.1 *Universal Variable Length Coding (UVLC)*

In the default entropy coding mode, a universal VLC is used to code all syntax. The table of codewords may be written in the following compressed form.

$$
\begin{array}{c}
1 \\
0\ x_0\ 1 \\
0\ x_1\ 0\ x_0\ 1 \\
0\ x_2\ 0\ x_1\ 0\ x_0\ 1 \\
0\ x_3\ 0\ x_2\ 0\ x_1\ 0\ x_0\ 1 \\
\cdots\cdots\cdots\cdots\cdots
\end{array}
$$

where x_n take values 0 or 1. We will sometimes refer to a codeword with its length in bits (L) and INFO = x_n .. x_1 x_0 . Notice that the number of bits in INFO is L/2 (division by truncation). The codewords are numbered from 0 and upwards. The definition of the numbering is:

Code_number = $2^{L/2}$ + INFO -1 (L/2 use division with truncation. INFO = 0 when L = 1)

Some of the first code numbers and codewords are written explicitly in the table below. As an example, for the code number 5, L = 5 and INFO = 10 (binary) = 2 (decimal)

Code number	Codewords in explicit form
0	1
1	0 0 1
2	0 1 1
3	0 0 0 0 1
4	0 0 0 1 1
5	0 1 0 0 1
6	0 1 0 1 1
7	0 0 0 0 0 0 1
8	0 0 0 0 0 1 1
9	0 0 0 1 0 0 1
10	0 0 0 1 0 1 1
11	0 1 0 0 0 0 1
.....	. . . .

When L and INFO is known, the regular structure of the table makes it easy to create a codeword bit by bit. Similarly, a decoder may easily read bit by bit until the last "1" which gives the end of the codeword. L and INFO is then readily available. For each parameter to be coded, there is a conversion rule from the parameter value to the code number (or L and INFO). **Error! Reference source not found.** lists the connection between code number and most of the parameters used in the present coding method.

Table 2: Connection between codeword number and parameter values.

Code_ number	RUN	MB_Type		Intra_pred_mode[1]		MVD	CBP		Tcoeff_chroma_DC[2]		Tcoeff_chroma_AC[2] Tcoeff_luma[2] Simple scan		Tcoeff_luma[2] Double scan	
		Intra	Inter	Prob0	Prob1		Intra	Inter	Level	Run	Level	Run	Level	Run
0	0	Intra4x4	16x16	0	0	0	47	0	EOB	-	EOB	-	EOB	-
1	1	0,0,0[3]	16x8	1	0	1	31	16	1	0	1	0	1	0
2	2	1,0,0	8x16	0	1	-1	15	1	-1	0	-1	0	-1	0
3	3	2,0,0	8x8	0	2	2	0	2	2	0	1	1	1	1
4	4	3,0,0	8x4	1	1	-2	23	4	-2	0	-1	1	-1	1
5	5	0,1,0	4x8	2	0	3	27	8	1	1	1	2	2	0
6	6	1,1,0	4x4	3	0	-3	29	32	-1	1	-1	2	-2	0
7	7	2,1,0	Intra4x4	2	1	4	30	3	3	0	2	0	1	2
8	8	3,1,0	0,0,0[3]	1	2	-4	7	5	-3	0	-2	0	-1	2
9	9	0,2,0	1,0,0	0	3	5	11	10	2	1	1	3	3	0
10	10	1,2,0	2,0,0	0	4	-5	13	12	-2	1	-1	3	-3	0
11	11	2,2,0	3,0,0	1	3	6	14	15	1	2	1	4	4	0
12	12	3,2,0	0,1,0	2	2	-6	39	47	-1	2	-1	4	-4	0
13	13	0,0,1	1,1,0	3	1	7	43	7	1	3	1	5	5	0
14	14	1,0,1	2,1,0	4	0	-7	45	11	-1	3	-1	5	-5	0
15	15	2,0,1	3,1,0	5	0	8	46	13	4	0	3	0	1	3
16	16	3,0,1	0,2,0	4	1	-8	16	14	-4	0	-3	0	-1	3
17	17	0,1,1	1,2,0	3	2	9	3	6	3	1	2	1	1	4
18	18	1,1,1	2,2,0	2	3	-9	51	9	-3	1	-2	1	-1	4
19	19	2,1,1	3,2,0	1	4	10	10	31	2	2	2	2	2	1
20	20	3,1,1	0,0,1	0	5	-10	12	35	-2	2	-2	2	-2	1
21	21	0,2,1	1,0,1	1	5	11	19	37	2	3	1	6	3	1
22	22	1,2,1	2,0,1	2	4	-11	21	42	-2	3	-1	6	-3	1
23	23	2,2,1	3,0,1	3	3	12	26	44	5	0	1	7	6	0
24	24	3,2,1	0,1,1	4	2	-12	28	33	-5	0	-1	7	-6	0
25	25		1,1,1	5	1	13	35	34	4	1	1	8	7	0
26	26		2,1,1	5	2	-13	37	36	-4	1	-1	8	-7	0
27	27		3,1,1	4	3	14	42	40	3	2	1	9	8	0
28	28		0,2,1	3	4	-14	44	39	-3	2	-1	9	-8	0
29	29		1,2,1	2	5	15	1	43	3	3	4	0	9	0
30	30		2,2,1	3	5	-15	2	45	-3	3	-4	0	-9	0
31	31		3,2,1	4	4	16	4	46	6	0	5	0	10	0
32	32			5	3	-16	8	17	-6	0	-5	0	-10	0
33	33			5	4	17	17	18	5	1	3	1	4	1
34	34			4	5	-17	18	20	-5	1	-3	1	-4	1
35	35			5	5	18	20	24	4	2	3	2	2	2
36	36					-18	24	19	-4	2	-3	2	-2	2
37	37					19	6	21	4	3	2	3	2	3
38	38					-19	9	26	-4	3	-2	3	-2	3
39	39					20	22	28	7	0	2	4	2	4
40	40					-20	25	23	-7	0	-2	4	-2	4
41	41					21	32	27	6	1	2	5	2	5
42	42					-21	33	29	-6	1	-2	5	-2	5
43	43					22	34	30	5	2	2	6	2	6
44	44					-22	36	22	-5	2	-2	6	-2	6
45	45					23	40	25	5	3	2	7	2	7
46	46					-23	38	38	-5	3	-2	7	-2	7
47	47					24	41	41	8	0	2	8	11	0
	..					..			..	..	..	..	..	..

[1] Prob0 and Prob1 defines the Intra prediction modes of two blocks relative to the prediction of prediction modes (see details in the section for Intra coding).

[2] For the entries above the horizontal line, the table is needed for relation between code number and Level/Run/EOB. For the remaining Level/Run combination there is a simple rule. The Level/Run combinations are assigned a code number according to the following priority: 1) sign of Level (+ -) 2) Run (ascending) 3) absolute value of Level (ascending).

[3] 16x16 based intra mode. The 3 numbers refer to values for (Imode,AC,nc) - see 3.5.9.3.

5.2 *Context-based Adaptive Binary Arithmetic Coding (CABAC)*

5.2.1 Overview

The entropy coding method of context-based adaptive binary arithmetic coding (CABAC) has three distinct elements compared to the default entropy coding method using a fixed, universal table of variable length codes (UVLC):

1. *Context modeling* provides estimates of conditional probabilities of the coding symbols. Utilizing suitable context models, given inter-symbol redundancy can be exploited by switching between different probability models according to already coded symbols in the neighborhood of the current symbol to encode.
2. *Arithmetic codes* permit non-integer number of bits to be assigned to each symbol of the alphabet. Thus the symbols can be coded almost at their entropy rate. This is extremely beneficial for symbol probabilities much greater than 0.5, which often occur with efficient context modeling. In this case, a variable length code has to spend at least one bit in contrast to arithmetic codes, which may use a fraction of one bit.
3. *Adaptive* arithmetic codes permit the entropy coder to adapt itself to non-stationary symbol statistics. For instance, the statistics of motion vector magnitudes vary over space and time as well as for different sequences and bit-rates. Hence, an adaptive model taking into account the cumulative probabilities of already coded motion vectors leads to a better fit of the arithmetic codes to the current symbol statistics.

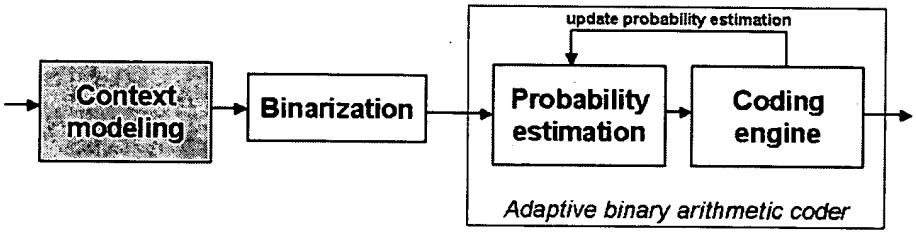


Figure 6: Generic block diagram of CABAC entropy coding scheme

Next we give a short overview of the main coding elements of the CABAC entropy coding scheme as depicted in Figure 6. Suppose a symbol related to an arbitrary syntax element is given, then, in a first step, a suitable model is chosen according to a set of past observations. This process of constructing a model conditioned on neighboring symbols is commonly referred to as *context modeling* and is the first step in the entropy coding scheme. The particular context models that are designed for each given syntax model are described in detail in Section 5.2.2 and Section 5.2.3. If a given symbol is non-binary valued, it will be mapped onto a sequence of binary decisions, so-called *bins*, in a second step. The actual *binarization* is done according to a given binary tree, as specified in Section 5.2.4. Finally, each binary decision is encoded with the *adaptive binary arithmetic coding* (AC) engine using the *probability estimates*, which have been provided either by the context modeling stage or by the binarization process itself. The provided models serve as a probability estimation of the related bins. After encoding of each bin, the related model will be updated with the encoded binary symbol. Hence, the model keeps track of the actual statistics.

5.2.2 Context Modeling for Coding of Motion and Mode Information

In this section we describe in detail the context modeling of our adaptive coding method for the syntax elements macroblock type (MB_type), motion vector data (MVD) and reference frame parameter (Ref_frame).

5.2.2.1 Context Models for Macroblock Type

We distinguish between MB_type for intra and inter frames. In the following, we give a description of the context models which have been designed for coding of the MB_type information in both cases. The subsequent process of mapping a non-binary valued MB_type symbol to a binary sequence in the case of inter frames will be given in detail in section 5.2.4.

5.2.2.1.1 Intra Pictures

For intra pictures, there are two possible modes for each macroblock, i.e. Intra4x4 and Intra16x16, so that signalling the mode information is reduced to transmitting a binary decision. Coding of this binary decision for a given macroblock is performed by means of context-based arithmetic coding, where the context of a current MB_type C is build by using the MB_types A and B[1] of neighboring macroblocks (as depicted in Figure 7) which are located in the causal past of the current coding event C. Since A and B are binary decisions, we define the actual context number *ctx_mb_type_intra(C)* of C by *ctx_mb_type_intra(C) = A + 2*B*, which results in four different contexts according to the 4 possible combinations of MB_type states for A and B.

In the case of MB_type Intra16x16, there are three additional parameters related to the chosen intra prediction mode, the occurrence of significant AC-coefficients and the coded block pattern for the chrominance coefficients, which have to be signaled. In contrast to the current test model, this information is not included in the mode information, but is coded separately by using distinct models as described in Section 5.2.3.

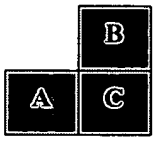
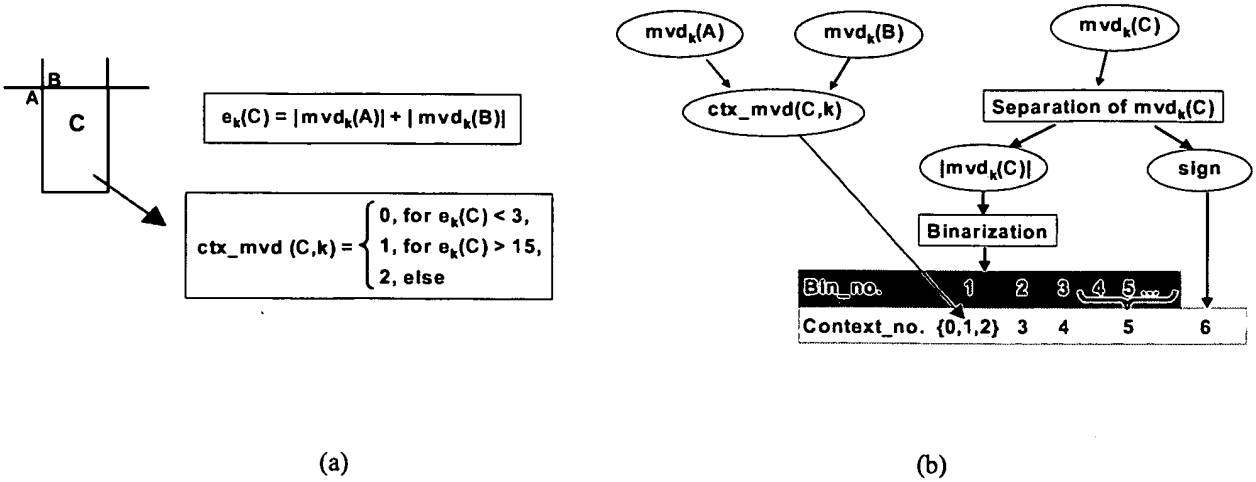


Figure 7: Neighboring symbols A and B used for conditional coding of a current symbol C.

5.2.2.1.2 P- and B-Pictures

Currently there a 10 different macroblock types for P-frames and 18 different macroblock types for B-frames, provided that the additional information of the 16x16 Intra mode is not considered as part of the mode information. Coding of a given MB_type information C is done similar to the case of intra frames by using a context model which involves the MB_type information A and B of previously encoded (or decoded) macroblocks (cp. Figure 7). However, here we only use the information whether the neighboring macroblocks of the given macroblock are of type Skip (P-frame) or Direct (B-frame), such that the actual context number *ctx_mb_type_inter(C)* is given in C-style notation for P-frame coding by *ctx_mb_type_inter(C) = ((A==Skip)?0:1) + 2*((B==Skip)?0:1)* and by *ctx_mb_type_inter(C) = ((A==Direct)?0:1) + 2*((B==Direct)?0:1)* for B-frame coding. Thus, we obtain 4 different contexts, which, however, are only used for coding of the first bin of the binarization ***b(C)*** of C, where the actual binarization of C will be performed as outlined in Section 5.2.4. For coding the second bin, a separate model is provided and for all remaining bins of ***b(C)*** two additional models are used as further explained in Sect. 5.2.3. Thus, a total number of 7 different models are supplied for coding of macroblock type information relating to P-and B-frames.

[1] For mathematical convenience the meaning of the variables A, B and C is context dependent.



(a) (b)

Figure 8: Illustration of the encoding process for a given residual motion vector component mvdk(C) of a block C: (a) Context selection rule. (b) Separation of mvdk(C) into sign and magnitude, binarization of the magnitude and assignment of context models to bin_nos.

5.2.2.2 Context Models for Motion Vector Data

Motion vector data consists of residual vectors obtained by applying motion vector prediction. Thus, it is a reasonable approach to build a model conditioned on the local prediction error. A simple measure of the local prediction error at a given block C is given by evaluating the L1-norm $e_k(A,B) = |mvd_k(A)| + |mvd_k(B)|$ of two neighboring motion vector prediction residues $mvd_k(A)$ and $mvd_k(B)$ for each component of a motion vector residue $mvd_k(C)$ of a given block, where A and B are neighboring blocks of block C, as shown in Figure 8 (a). If one of the neighboring blocks belongs to an adjacent macroblock, we take the residual vector component of the leftmost neighboring block in the case of the upper block B, and in the case of the left neighboring block A we use the topmost neighboring block. If one of the neighboring blocks is not available, because, for instance, the current block is at the picture boundary, we discard the corresponding part of e_k. By using e_k, we now define a context model *ctx_mvd(C,k)* for the residual motion vector component $mvd_k(C)$ consisting of three different context models:

$$ctx\_mvd(C,k) = \begin{cases} 0, e_k(C) < 3, \\ 1, e_k(C) > 15, \\ 2, else. \end{cases}$$

For the actual coding process, we separate $mvd_k(C)$ in sign and modulus, where only the first bin of the binarization of the modulus $|mvd_k(C)|$ is coded using the context models *ctx_mvd(C,k)*. For the remaining bins, we have three additional models: two for the second and the third bin and a third model for all remaining bins. In addition, the sign coding routine is provided with a separate model. This results in a total sum of 7 different models for each vector component (see Figure 8).

In the case of B-frame coding, an additional syntax element has to be signaled when the bi-directional mode is chosen. This element represents the block size (Blk_size), which is chosen for forward or backward motion prediction. The related code number value ranges between 0 and 6 according to the 7 possible block shapes **Error! Reference source not found.**. Coding of Blk_size is done by using the binarization of the P_MB_type as described in Section 5.2.3.

5.2.2.3 Context Models for Reference Frame Parameter

If the option of temporal prediction from more than one reference frame is enabled, the chosen reference frame for each macroblock must be signaled. Given a macroblock and its reference frame parameter as a symbol C according to the definition in Sect. 3, a context model is built by using symbols A and B of the reference frame parameter belonging to the two neighboring macroblocks (cp. Figure 7). The actual context number of C is then defined by *ctx_ref_frame(C) = ((A==0)?0:1) + 2*((B==0)?0:1)*, such that *ctx_ref_frame(C)* indicates one of four models used for coding of the first bin of the binary equivalent

b(C) of C. Two additional models are given for the second bin and all remaining bins of *b(C)*, which sums up to a total number of six different models for the reference frame information.

5.2.3 Context Modeling for Coding of Texture Information

This section provides detailed information about the context models used for the syntax elements of coded block pattern (CBP), intra prediction mode (IPRED) and (RUN, LEVEL) information.

5.2.3.1 Context Models for Coded Block Pattern

Except for MB_type Intra16x16, the context modeling for the coded block pattern is treated as follows. There are 4 luminance CBP bits belonging to 4 8x8 blocks in a given macroblock. Let C denote such a Y-CBP bit, then we define *ctx_cbp_luma(C) = A + 2*B,* where A and B are Y-CBP bits of the neighboring 8x8 blocks, as depicted in Figure 7. The remaining 2 bits of CBP are related to the chrominance coefficients. In our coding approach, these bits are translated into two dependant binary decisions, such that, in a first step, we send a bit *cbp_chroma_sig* which signals whether there are significant chrominance coefficients at all. The related context model is of the same kind as that of the Y-CBP bits, i.e. *ctx_cbp_chroma_sig(C) = A + 2*B,* where A and B are now notations for the corresponding *cbp_chroma_sig* bits of neighboring macroblocks. If *cbp_chroma_sig = 1* (non-zero chroma coefficients exist), a second bit *cbp_chroma_ac* related to the significance of AC chrominance coefficients has to be signalled. This is done by using a context model conditioned on the *cbp_chroma_ac* decisions A and B of neighboring macroblocks, such that *ctx_cbp_chroma_AC(C) = A + 2*B.* Note, that due to the different statistics there are different models for Intra and Inter macroblocks, so that the total number of different models for CBP amounts to 2*3*4=24. For the case of MB_type Intra16x16, there are three additional models, one for the binary AC decision and two models for each of the two chrominance CBP bits.

5.2.3.2 Context Models for Intra Prediction Mode

In Intra4x4 mode, coding of the intra prediction mode C of a given block is conditioned on the intra prediction mode of the previous block A to the left of C (cp. Figure 7). In fact, it is not the prediction mode number itself which is signaled and which is used for conditioning but rather its predicted order similar as it is desribed in Sect. 3.5.8. There are 6 different prediction modes and for each mode, two different models are supplied: one for the first bin of the binary equivalent of C and the other for all remaining bins. Together with two additional models for the two bits of the prediction modes of MB_type Intra16x16 (in binary representation), a total number of 14 different models for coding of intra prediction modes is given.

5.2.3.3 Context Models for Run/Level

Coding of (RUN, LEVEL) pairs is conditioned on the scanning mode, the DC/AC block type, the luminance/chrominance, and the intra/inter macroblock decision. Thus, a total number of 9 different block types are given according to Table 3, which, in turn, results in 9 different contexts. In contrast to the current test model, RUN and LEVEL are coded separately in our coding approach, as described in the following two subsections.

Ctx_run_level	Block Type
0	Double Scan
1	Single Scan, Inter
2	Single Scan, Intra
3	Intra16x16, DC
4	Intra16x16, AC
5	Chroma, DC, Inter
6	Chroma, DC, Intra
7	Chroma, AC, Inter
8	Chroma, AC, Intra

Table 3: Numbering of the different context models used for coding of RUN and LEVEL

5.2.3.3.1 Context-based Coding of LEVEL Information

For a given block C, the LEVEL information is first separated into sign and magnitude. According to its context *ctx_run_level(C)* four different models are chosen, where one model is used for coding of the sign information and the remaining 3 models are used for the first, the second and all remaining bins of the binarization of LEVEL. If LEVEL≠0 (EOB), the corresponding RUN is coded in a subsequent coding routine, as described in the following section.

5.2.3.3.2 Context-based Coding of RUN Information

For each context *ctx_run_level(C)* two separate models are provided for the coding of RUN; one model for the first bin and the second model for all remaining bins of the binary sequence related to RUN.

Code symbol	Binarization							
0	1							
1	0	1						
2	0	0	1					
3	0	0	0	1				
4	0	0	0	0	1			
5	0	0	0	0	0	1		
6	0	0	0	0	0	0	1	
. . .	.	.	.	.	.	.	.	.
Bin_no.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	..

Table 4: Binarization by means of the unary code tree

5.2.4 Binarization of Non-Binary Valued Symbols

A non-binary valued symbol will be decomposed into a sequence of binary decisions. Except for the MB_type syntax element we use the binarization given by the unary code tree in Table 4.

P_MB_type	Binarization
0	0
1	1 0 0
2	1 0 1
3	1 1 0 0 0
4	1 1 0 0 1
5	1 1 0 1 0
6	1 1 0 1 1
7	1 1 1 0 0
8	1 1 1 0 1
9	1 1 1 1 0
Bin_no	**1 2 3 4 5**

(a)

B_MB_type	Binarization
0	0
1	1 0 0
2	1 0 1
3	1 1 0 0 0
4	1 1 0 0 1
5	1 1 0 1 0
6	1 1 0 1 1
7	1 1 1 0 0 0 0
.	.
17	1 1 1 1 0 1 0
Bin_no	**1 2 3 4 5 6 7**

(b)

Table 5: (a) Binarization for P-frame MB_type and (b) for B-frame MB_type

For the binary decomposition of the MB_type symbols of P- or B-frames, which are of limited range (0 ... 9) or (0 ... 17) respectively, an alternative binarization is used, which is shown in Table 5.

5.2.5 Adaptive Binary Arithmetic Coding

At the beginning of the overall encoding of a given frame the probability models associated with all 126 different contexts are initialized with a pre-computed start distribution. For each symbol to encode the frequency count of the related binary decision is updated, thus providing a new probability estimate for the next coding decision. However, when the total number of occurrences of a given model exceeds a pre-defined threshold, the frequency counts will be scaled down. This periodical rescaling exponentially weighs down past observations and helps to adapt to the non-stationarity of a source. The binary arithmetic coding engine used in our presented approach is a straightforward implementation similar to that given in Witten et al., "Arithmetic Coding for Data Compression", *Comm. of the ACM*, 30 (6), 1987, pp.520-541.

6 Test model issues

6.1 Motion Estimation and Mode Decision

6.1.1 Low-complexity mode

6.1.1.1 Finding optimum prediction mode

Both for intra prediction and motion compensated prediction, a similar loop as indicated in Figure 9 is run through. The different elements will be described below.

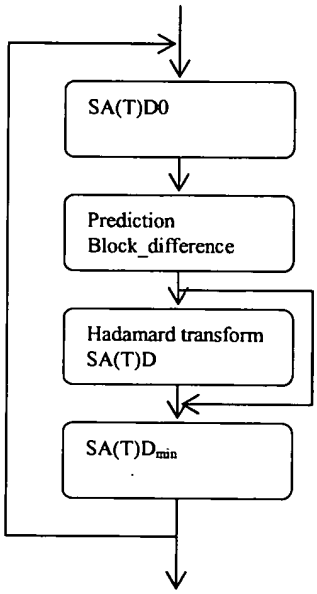


Figure 9 Loop for prediction mode decision

6.1.1.1.1 SA(T)D0

The SA(T)D to be minimised is given a 'bias' value initially in order to favour prediction modes that need few bits to be signalled. This bias is basically a parameter representing bit usage times $QP_0(QP)$

Intra mode decision: SA(T)D0 = $QP_0(QP)$xOrder_of_prediction_mode (see above)

Motion vector search: SA(T)D0 = $QP_0(QP)$x(Bits_to_code_vector + 2xcode_number_of_ref_frame)

In addition there are two special cases:

- For motion prediction of a 16x16 block with 0 vector components, 16x$QP_0(QP)$ is subtracted from SA(T)D to favour the skip mode.
- For the whole intra macroblock, 24x$QP_0(QP)$ is added to the SA(T)D before comparison with the best SA(T)D for inter prediction. This is an empirical value to prevent using too many intra blocks.

6.1.1.1.2 Block_difference

For the whole block the difference between the original and prediction is produced

Diff(i,j) = Original(i,j) - Prediction(i,j)

6.1.1.1.3 Hadamard transform

For integer pixel search (see below) we use SAD based on Diff(i,j) for decision. Hence no Hadamard is done and we use SAD instead of SATD.

$$SAD = \sum_{i,j} |Diff(i,j)|$$

However, since we will do a transform of Diff(i,j) before transmission, we will do a better optimisation if a transform is done before producing SAD. Therefore a two dimensional transform is performed in the decision loop for selecting intra modes and for fractional pixel search (see below). To simplify implementation, the Hadamard transform is chosen in this mode decision loop. The relation between pixels and basis vectors (BV) in a 4 point Hadamard transform is illustrated below (not normalized):

	Pixels →			
B	1	1	1	1
V	1	1	-1	-1
↓	1	-1	-1	1
	1	-1	1	-1

This transformation is performed horizontally and vertically and result in DiffT(i,j). Finally SATD for the block and for the present prediction mode is produced.

$$SATD = (\sum_{i,j} |DiffT(i,j)|)/2$$

Choose the prediction mode that results in the minimum $SA(T)D_{min}$.

6.1.1.2 Encoding on macroblock level

6.1.1.2.1 Intra coding

When starting to code a macroblock, intra mode is checked first. For each 4x4 block, full coding indicated in Figure 9 is performed. At the end of this loop the complete macroblock is intra coded and a $SATD_{intra}$ is calculated.

6.1.1.2.2 Table for intra prediction modes to be used at the encoder side

Table 6 gives the table of intra prediction modes according to probability of each mode to be used on the decoder side. On the encoder side we need a sort of inverse table. Prediction modes for A and B are known as in Table 1. For the encoder we have found a Mode that we want to signal with an ordering number in the bitstream (whereas on the decoder we receive the order in the bitstream and want to convert this to a mode). Table 6 is therefore the relevant table for the encoder. Example: Prediction mode for A and B is 2. The string in Table 6 is 2 1 0 3 4 5. This indicates that prediction mode 0 has order 2 (third most probable). Prediction mode 1 is second most probable and prediction mode 2 has order 0 (most probable) etc. As in Table 1 '–' indicates that this instance can not occur because A or B or both are outside the picture.

Table 6 Prediction ordering to be used in the bitstream as a function of prediction mode (see text).

B\A	outside	0	1	2	3	4	5
outside	0-----	021---	102---	120---	012---	012---	012---
0	0---12	025314	104325	240135	143025	035214	045213
1	0---12	014325	102435	13 0245	032145	024315	015324
2	0---12	012345	102345	210345	132045	032415	013245
3	0---12	135024	214035	320154	143025	145203	145032
4	1---02	145203	125403	250314	245103	145203	145302
5	1---20	245310	015432	120534	245130	245301	135420

6.1.1.2.3 Inter mode selection

Next motion vector search is performed for motion compensated prediction. A search is made for all 7 possible block structures for prediction as well as from the 5 past decoded pictures. This result in 35 combinations of blocksizes and reference frames.

6.1.1.2.4 Integer pixel search

The search positions are organised in a 'spiral' structure around the predicted vector (see vector prediction). The numbering from 0 and upwords for the first positions are listed below:

```
. .  .  .  .  .
.15  9 11 13 16
.17  3  1  4 18
.19  5  0  6 20
.21  7  2  8 22
.23 10 12 14 24
```

A parameter MC_range is used as input for each sequence. To speed up the search process, the search range is further reduced:

- Search range is reduced to: Range = MC_range/2 for all block sizes except 16x16 in prediction from the most recent decoded picture.
- The range is further reduced to: Range = Range/2 for search relative to all older pictures.

After Range has been found, the centre for the spiral search is adjusted so that:

- No vector position is outside the picture.
- The (0,0) vector position is within the search range. This is done by clipping the horizontal and vertical positions of the search centre to ±Range.

6.1.1.2.5 Fractional pixel search

Fractional pixel search is performed in two steps. This is illustrated below where capital letters represent integer positions, numbers represent ½ pixel positions and lower case letters represent ¼ pixel positions.

```
A       B       C

    1   2   3

D   4   E   5   F
      a b c
    6 d 7 e 8
      f g h
G       H       I
```

Assume that the integer search points to position E. Then ½ pixel positions 1,2,3,4,5,6,7,8 are searched. Assume that 7 is the best position. Then the ¼ pixel positions a,b,c,d,e,f,g,h are searched. (Notice that by this procedure a position with 'more low pass filtering' – see 3.7.1 - is automatically checked).

After fractional pixel search has been performed for the complete macroblock, the SATD for the whole macroblock is computed: $SATD_{inter}$.

6.1.1.2.6 Decision between intra and inter

If $SATD_{intra} < SATD_{inter}$ intra coding is used. Otherwise inter coding is used.

6.1.2 High-complexity mode

6.1.2.1 Motion Estimation

For each block or macroblock the motion vector is determined by full search on integer-pixel positions followed by sub-pixel refinement.

6.1.2.1.1 Integer-pixel search

As in low-complexity mode, the search positions are organized in a spiral structure around a prediction vector. The full search range given by MC_range is used for all INTER-modes and reference frames. To

speed up the search process , the prediction vector of the 16x16 block is used as center of the spiral search for all INTER-modes. Thus the SAD values for 4x4 blocks can be pre-calculated for all motion vectors of the search range and than used for fast SAD calculation of all larger blocks. The search range is not forced to contain the (0,0)-vector.

6.1.2.1.2 Fractional pixel search

The fractional pixel search is performed as in the low-complexity case.

6.1.2.1.3 Finding the best motion vector

The integer-pixel motion search as well as the sub-pixel refinement returns the motion vector that minimizes

$$J(\mathbf{m}, \lambda_{MOTION}) = SA(T)D(s, c(\mathbf{m})) + \lambda_{MOTION} \cdot R(\mathbf{m} - \mathbf{p})$$

with $\mathbf{m} = (m_x, m_y)^T$ being the motion vector, $\mathbf{p} = (p_x, p_y)^T$ being the prediction for the motion vector, and λ_{MOTION} being the Lagrange multiplier. The rate term $R(\mathbf{m} - \mathbf{p})$ represents the motion information only and is computed by a table-lookup. The rate is estimated by using the universal variable length code (UVLC) table, even if CABAC is used as entropy coding method. For integer-pixel search, SAD is used as distortion measure. It is computed as

$$SAD(s,\ c(\mathbf{m})) = \sum_{x=1,\ y=1}^{B,\ B} \left| s[x, y] - c[x - m_x, y - m_y] \right|, \quad B = 16, 8 \text{ or } 4 .$$

with *s* being the original video signal and *c* being the coded video signal. In the sub-pixel refinement search, the distortion measure SATD is calculated after a Hadamard transform (see section 6.1.1.1.3). The Lagrangian multiplier λ_{MOTION} is given by

$$\lambda_{MOTION,P} = \sqrt{5} \cdot e^{QP/20} \cdot \sqrt{\frac{QP+5}{34-QP}}$$

P-frames and

$$\lambda_{MOTION,B} = 2 \cdot \sqrt{5} \cdot e^{QP/20} \cdot \sqrt{\frac{QP+5}{34-QP}}$$

for B-frames, where QP is the macroblock quantization parameter.

6.1.2.1.4 Finding the best reference frame

The determination of the reference frame *REF* and the associated motion vectors for the *NxM* inter modes in P-frames and the *FWD NxM* modes in B-frames is done after motion estimation by minimizing

$$J(REF \mid \lambda_{MOTION}) = SATD(s, c(REF, \mathbf{m}(REF))) + \lambda_{MOTION} \cdot (R(\mathbf{m}(REF) - \mathbf{p}(REF)) + R(REF)) ;$$

The rate term $R(REF)$ represents the number of bits associated with choosing REF and is computed by table-lookup using UVLC. The reference frame and block sizes for the bi-directional mode are chosen as combination of the "best" forward and backward mode.

6.1.2.2 Mode decision

6.1.2.2.1 Macroblock mode decision

The macroblock mode decision is done by minimizing the Lagrangian functional

$$J(s,c,MODE \mid QP,\lambda_{MODE}) = SSD(s,c,MODE \mid QP) + \lambda_{MODE} \cdot R(s,c,MODE \mid QP)$$

where QP is the macroblock quantizer, λ_{MODE} is the Lagrange multiplier for mode decision, and $MODE$ indicates a mode chosen from the set of potential prediction:

I-frame: $MODE \in \{INTRA4x4,\ INTRA16x16\}$,

P-frame: $MODE \in \begin{Bmatrix} INTRA4x4,\ INTRA16x16,\ SKIP, \\ 16x16,\ 16x8,\ 8x16,\ 8x8,\ 8x4,\ 4x8,\ 4x4 \end{Bmatrix}$,

B-frame: $MODE \in \begin{Bmatrix} INTRA4x4,\ INTRA16x16,\ BIDIRECT,\ DIRECT, \\ FWD16x16,\ FWD16x8,\ FWD8x16,\ FWD8x8,\ FWD8x4, \\ FWD4x8,\ FWD4x4,\ BAK16x16,\ BAK16x8,\ BAK8x16, \\ BAK8x8,\ BAK8x4,\ BAK4x8,\ BAK4x4 \end{Bmatrix}$.

Note that the $SKIP$ mode refers to the 16x16 mode where no motion and residual information is encoded. SSD is the sum of the squared differences between the original block s and its reconstruction c being given as

$$SSD(s,c,MODE \mid QP) = \sum_{x=1,y=1}^{16,16} \left(s_Y[x,y] - c_Y[x,y,MODE \mid QP]\right)^2 + \sum_{x=1,y=1}^{8,8} \left(s_U[x,y] - c_U[x,y,MODE \mid QP]\right)^2 + \sum_{x=1,y=1}^{8,8} \left(s_V[x,y] - c_V[x,y,MODE \mid QP]\right)^2,$$

and $R(s,c,MODE \mid QP)$ is the number of bits associated with choosing $MODE$ and QP including the bits for the macroblock header, the motion, and all DCT blocks. $c_Y[x,y,MODE \mid QP]$ and $s_Y[x,y]$ represent the reconstructed and original luminance values; c_U, c_V and s_U, s_V the corresponding chrominance values.

The Lagrangian multiplier λ_{MODE} is given by

$$\lambda_{MODE,P} = 5 \cdot e^{QP/10} \cdot \left(\frac{QP+5}{34-QP}\right)$$

for I- and P-frames and

$$\lambda_{MODE,B} = 20 \cdot e^{QP/10} \cdot \left(\frac{QP+5}{34-QP}\right)$$

for B-frames, where QP is the macroblock quantization parameter.

6.1.2.2.2 INTER 16x16 mode decision

The *INTER16x16* mode decision is performed by choosing the INTER16x16 mode which results in the minimum SATD value.

6.1.2.2.3 INTER 4x4 mode decision

For the *INTRA4x4* prediction, the mode decision for each 4x4 block is performed similar to the macroblock mode decision by minimizing

$$J(s,c,IMODE \mid QP,\lambda_{MODE}) = SSD(s,c,IMODE \mid QP) + \lambda_{MODE} \cdot R(s,c,IMODE \mid QP)$$

where QP is the macroblock quantizer, λ_{MODE} is the Lagrange multiplier for mode decision, and $IMODE$ indicates an intra prediction mode:

$$IMODE \in \{DC, HOR, VERT, DIAG, DIAG\_RL, DIAG\_LR\}.$$

SSD is the sum of the squared differences between the original 4x4 block luminance signal$s$ and its reconstruction *c*, and $R(s,c,IMODE \mid QP)$ represents the number of bits associated with choosing *IMODE*. It includes the bits for the intra prediction mode and the DCT-coefficients for the 4x4 luminance block. The rate term is computed using the UVLC entropy coding, even if CABAC is used for entropy coding.

6.1.2.3 Algorithm for motion estimation and mode decision

The procedure to encode one macroblock s in a I-, P- or B-frame in the high-complexity mode is summarized as follows.

1. Given the last decoded frames, λ_{MODE}, λ_{MOTION}, and the macroblock quantizer QP
2. Choose intra prediction modes for the *INTRA 4x4* macroblock mode by minimizing

$$J(s,c,IMODE \mid QP,\lambda_{MODE}) = SSD(s,c,IMODE \mid QP) + \lambda_{MODE} \cdot R(s,c,IMODE \mid QP)$$

   with $IMODE \in \{DC, HOR, VERT, DIAG, DIAG\_RL, DIAG\_LR\}$.

3. Determine the best $INTRA16x16$ prediction mode by choosing the mode which results in the minimum SATD.

4. Perform motion estimation and reference frame selection by minimizing

$$J(REF, \mathbf{m}(REF) \mid \lambda_{MOTION}) = SA(T)D(s, c(REF, \mathbf{m}(REF))) + \lambda_{MOTION} \cdot (R(\mathbf{m}(REF) - \mathbf{p}(REF)) + R(REF))$$

for each reference frame and motion vector of a possible macroblock mode.

5. Choose the macroblock prediction mode by minimizing

$$J(s, c, MODE \mid QP, \lambda_{MODE}) = SSD(s, c, MODE \mid QP) + \lambda_{MODE} \cdot R(s, c, MODE \mid QP),$$

given QP and λ_{MODE} when varying $MODE$. $MODE$ indicates a mode out of the set of potential macroblock modes:

I-frame: $MODE \in \{INTRA4x4, INTRA16x16\}$,

P-frame: $MODE \in \begin{Bmatrix} INTRA4x4, INTRA16x16, SKIP, \\ 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4 \end{Bmatrix}$,

B-frame: $MODE \in \begin{Bmatrix} INTRA4x4, INTRA16x16, BIDIRECT, DIRECT, \\ FWD16x16, FWD16x8, FWD8x16, FWD8x8, FWD8x4, \\ FWD4x8, FWD4x4, BAK16x16, BAK16x8, BAK8x16, \\ BAK8x8, BAK8x4, BAK4x8, BAK4x4 \end{Bmatrix}$.

The computation of $J(s, c, SKIP \mid QP, \lambda_{MODE})$ and $J(s, c, DIRECT \mid QP, \lambda_{MODE})$ is simple. The costs for the other macroblock modes are computed using the intra prediction modes or motion vectors and reference frames, which have been estimated in steps 2- 4.

6.2 Quantization

For each transform coefficient K the quantized LEVEL is produced in the following way:

LEVEL = (KxA(QP) + fx2^{20})/2^{20} |f| is 1/3 for intra and 1/6 for inter blocks and f has the same sign as K.

6.3 Elimination of single coefficients in inter macroblocks

6.3.1 Luma

With the small 4x4 blocks, it may happen that for instance a macroblock has only one nonzero coefficient with |Level| =1. This will probably be a very expensive coefficient and it could have been better to set it to zero. For that reason a procedure to check single coefficients have been implemented for inter luma blocks. During the quantization process, a parameter **Single_ctr** is accumulated depending on Run and Level according to the following rule:

If Level = 0 or (|Level| = 1 and Run > 5) nothing is added to **Single_ctr**.

If |Level| > 1, 9 is added to **Single_ctr**.

If |Level| = 1 and Run < 6, a value T(Run) is added to **Single_ctr**. where T(0:5) =(3,2,2,1,1,1)

If the accumulated **Single_ctr** for a 8x8 block is less than 4, all coefficients of that luma block are set to zero. Similarly, if the accumulated **Single_ctr** for the whole macroblock is less than 6, all coefficients of that luma macroblock are set to zero.

6.3.2 Chroma

A similar method to the one for luma is used. **Single_ctr** is calculated similarly for each chroma component, but for AC coefficients only and for the whole macroblock.

If the accumulated **Single_ctr** for each chroma component of a macroblock is less than 7, all the AC chroma coefficients of that component for the whole macroblock are set to zero.

7 B-pictures

7.1 Introduction

This is a first definition of B pictures to be used in TML. It is mainly intended to get started on work of testing relevant coding tools. Due to the early stage of definition, a separate description and definition of syntax elements is included in this section. In a later version of TML it is foreseen that B-frames will be fully incorporated in the remaining definition. The use of B pictures is indicated in PTYPE.

Temporal scalability is achieved using bi-directionally predicted pictures, or B pictures. The B pictures are predicted from either or both the previous and subsequent reconstructed pictures to achieve improved coding efficiency as compared to that of P pictures. The B pictures are disposable, since the B pictures are not used as reference pictures for the prediction of any other pictures. This property allows B pictures to be discarded without destroying the ability to decode the sequence and adversely affecting the quality of any subsequent pictures, thus providing temporal scalability. Figure A.1 illustrates the predictive structure with two B pictures inserted between I/P pictures.

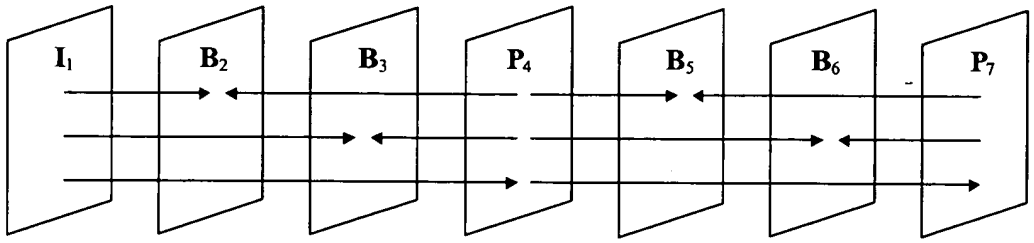


FIGURE A.1/H.26L

The location of B pictures in the bitstream is in a data-dependence order rather than in temporal order. Pictures that are dependent on other pictures shall be located in the bitstream after the pictures on which they depend. For example, as illustrated in Figure A.1, B_2 and B_3 are dependent on I_1 and P_4, and B_5 and B_6 are dependent on P_4 and P_7. Therefore the bitstream syntax order of the encoded pictures would be I_1, P_4, B_2, B_3, P_7, B_5, B_6, ... However, the display order of the decoded pictures should be I_1, B_2, B_3, P_4, B_5, B_6, P_7, ... The difference between the bitstream order of encoded pictures and the display order of decoded pictures will increase latency and memory to buffer the P pictures.

There is no limit to the number of B pictures that may be inserted between each I/P picture pair. The maximum number of such pictures may be signaled by external means (for example Recommendation H.245). The picture height, width, and pixel aspect ratio of a B picture shall always be equal to those of its temporally subsequent reference picture.

The B pictures described in this section support multiple reference frame prediction. The maximum number of previous reference frames that may be used for prediction in B pictures must be less than or equal to the number of reference frames used in the immediately following P frame, and it may be signaled by external means (for example Recommendation H.245). The use of this mode is indicated by PTYPE.

7.2 Five Prediction modes

There are five different prediction modes supported by B pictures. They are direct, forward, backward, bi-directional and the intra prediction modes. Both direct mode and bi-directional mode are bi-directional prediction. The only difference is that the bi-directional mode uses separate motion vectors for forward and backward prediction, whereas the forward and backward motion vectors of the direct mode are derived from the motion vectors used in the corresponding macroblocks of the subsequent reference frame. In the direct mode, the same number of motion vectors are used as are used in the reference macroblock for prediction. To calculate prediction blocks for the direct and bi-directional prediction mode, the forward and backward motion vectors are used to obtain appropriate blocks from reference frames and then these blocks are averaged by dividing the sum of the two prediction blocks by two.

Forward prediction means prediction from a previous reference picture, and backward prediction means prediction from a temporally subsequent reference picture.

The intra prediction means to encode the macroblock by using intra coding.

7.3 Finding optimum prediction mode

SA(T)D is initialized by a bias value to favor a prediction mode that needs few bits to be transmitted. This bias value is bit usage times $QP_0(QP)$ for the given coding mode.

For flat regions having zero motion, B pictures basically fail to make effective use of zero motion and instead are penalized in performance by selecting 16x16 intra mode. Therefore, in order to prevent assigning 16x16 intra mode to a region with little details and zero motion, SA(T)D of direct mode is subtracted by $16xQP_0(QP)$ to bias the decision toward selecting the direct mode.

And SA(T)D of 4x4 intra mode employs the same manner as section 5.1.

The calculation of SA(T)D at each mode is as follows.

- Forward prediction mode :

  $SA(T)D0 = QP_0(QP) \times (2 \times code\_number\_of\_Ref\_frame + Bits\_to\_code\_MVDFW)$

- Backward prediction mode :

  $SA(T)D0 = QP_0(QP) \times Bits\_to\_code\_MVDBW$

- Bi-directional prediction mode :

  $SA(T)D0 = QP_0(QP) \times (2 \times code\_number\_of\_Ref\_frame + Bits\_to\_code\_forward\_Blk\_size + Bits\_to\_code\_backward\_Blk\_size + Bits\_to\_code\_MVDFW + Bits\_to\_code\_MVDBW)$

- Direct prediction mode :

  $SA(T)D = SA(T)D - 16 \times QP_0(QP)$

- 4x4 Intra mode :

  $SA(T)D0 = QP_0(QP) \times Order\_of\_prediction\_mode$

  $SA(T)D = SA(T)D + 24 \times QP_0(QP)$

Finally the mode with the minimum $SA(T)D_{min}$ is selected as an optimum prediction mode.

7.4 Syntax

Some additional syntax elements are needed for B pictures. The structure of B picture related fields is shown in Figure A.2. On the Ptype, two picture types shall be added to include B pictures with and without multiple reference frame prediction. On the MB_type, different macroblock types shall be defined to indicate the different prediction types for B pictures. The fields of Blk_size, MVDFW, and MVDBW shall be inserted to enable bi-directional prediction.

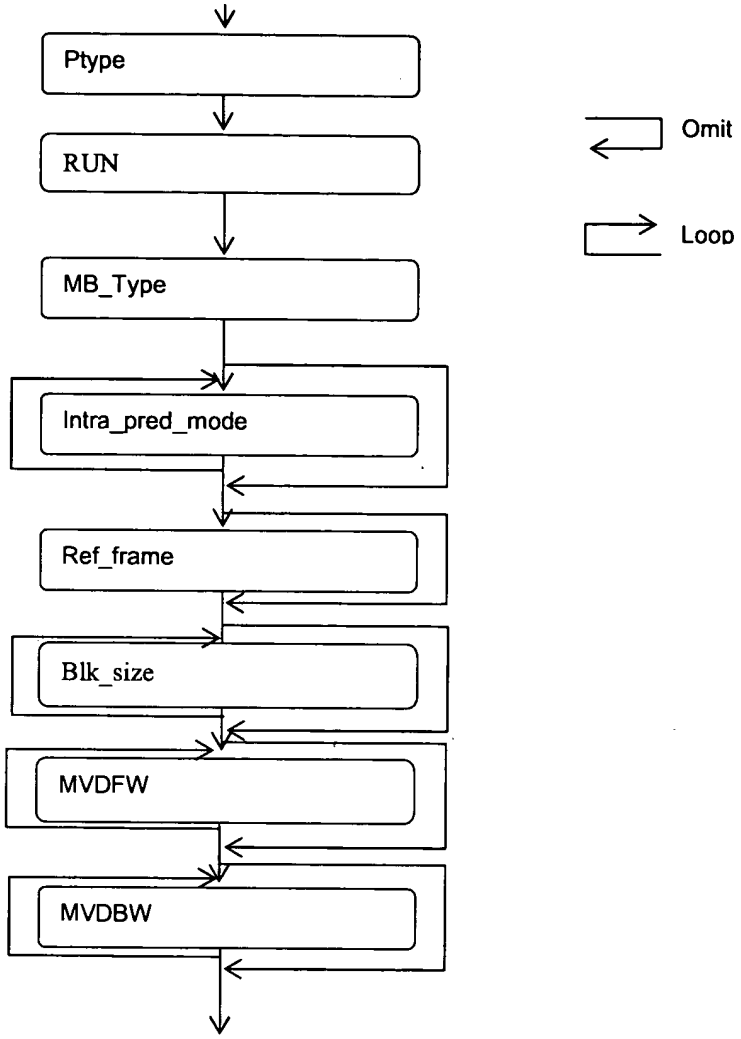


FIGURE A.2/H.26L

7.4.1 Picture type (Ptype) and RUN

See sections 3.2 and 3.3 for definitions.

7.4.2 Macro block type (MB_type)

The MB_type indicates the prediction mode and block size used to encode each macroblock. As mentioned earlier, five different prediction modes are supported by B pictures. For the forward, backward and bi-directional prediction modes, a macroblock is predicted from either or both of the previous and subsequent pictures with block size NxM. Table A.1 shows the macroblock types and the included data elements for B pictures.

In "Direct" prediction type, no motion vector data is transmitted.

The "Forward_NxM" indicates that the macroblock is prediction from a previous picture with block size NxM. The "backward_NxM" indicates that the macroblock is prediction from a subsequent picture with block size NxM. For each NxM block, motion vector data is provided. Therefore, depending on N and M, up to 16 sets of motion vector data have to be transmitted for a macroblock.

For the "Bi-directional" prediction type, the parameter Blk_size is used to indicate the block size used for forward and backward motion prediction (the Blk_size field is described in detail below). Both forward and backward motion vector data sets are transmitted. Depending on the block size indicated in Blk_size, up to 16 fields of motion vector data is transmitted for each of forward and backward prediction for a macroblock.

The "Intra_4x4" and "Intra_16x16" prediction type indicates that the macroblock is encoded by intra coding with different intra prediction modes which are defined in the same manner as section 3.4. No transmitted motion vector data is needed for intra mode.

7.4.3 Intra prediction mode (Intra_pred_mode)

As present, Intra_pred_mode indicates which intra prediction mode is used for a macroblock. Intra_pred_mode is present when Intra_4x4 prediction type is indicated in the MB_type. The code_number is same as that described in the Intra_pred_mode entry of Table 1.

7.4.4 Reference Frame (Ref_frame)

At present, Ref_frame indicates the position of the reference frame in the reference frame buffer to be used for forward motion compensated for current macroblock. Ref_frame is present only when the Ptype signals the use of multiple reference frames and only when the present MB_type indicates Forward_NxM or Bi-directional prediction type. Decoded I/P pictures are stored in the reference frame buffer in first-in-first-out manner and the most recently decoded I/P frame is always stored at position 0 in the reference frame buffer. The code_number for Ref_frame is described in Table A.2.

TABLE A.1/H.26L

MB_Type and related data elements for B pictures

0	Direct					
Code_number	Prediction Type	Intra_pred _mode	Ref_frame[1]	Blk_size	MVDFW	MVDBW
1	Forward_16x16		X		X	
2	Backward_16x16					X
3	Bi-directional		X	X	X	X
4	Forward_16x8		X		X	
5	Backward_16x8					X
6	Forward_8x16		X		X	
7	Backward_8x16					X
8	Forward_8x8		X		X	
9	Backward_8x8					X
10	Forward_8x4		X		X	
11	Backward_8x4					X
12	Forward_4x8		X		X	
13	Backward_4x8					X
14	Forward_4x4		X		X	
15	Backward_4x4					X
16	Intra_4x4	X				
17	Intra_16x16[2]					
...	...					

[1] Ref_frame is a valid field only when the usage of multiple reference frames is present in Ptype, e.g., when Ptype=4 the Ref_frame field is present.

[2] Intra_16x16 indicates 16x16 based intra mode and should represent 24 different prediction modes as defined in section 3.4.9 in Q15-J-28. For code_number greater than 16 in Table A.1, please see the code numbers from 9 and upwards in the field of inter MB_type of Table 1 in Q15-J-28 for reference.

TABLE A.2/H.26L

Code_number for ref_frame

Code_number	Reference frame
0	The most recent previous frame (1 frame back)
1	2 frames back
2	3 frames back
...	...

7.4.5 Block Size (Blk_size)

If present, Blk_size indicates which block size is used for forward and backward motion prediction in a macroblock as described in Table A.3. Blk_size is present only when Bi-directional prediction type is indicated in the MB_type. There are two sets of Blk_size data, one for forward motion vector data, and another for backward motion vector data.

TABLE A.3/H.26L

Code_number for Blk_size

Code_number	Block Size
0	1 16x16 block
1	4 8x8 blocks
2	2 16x8 blocks
3	2 8x16 blocks
4	2 8x4 blocks
5	8 4x8 blocks
6	16 4x4 blocks

7.4.6 Motion vector data (MVDFW, MVDBW)

MVDFW is the motion vector data for the forward vector, if present. MVDBW is the motion vector data for the backward vector, if present. If so indicated by MB_type or Blk_size (bi-directional prediction type only), vector data for 1-16 blocks are transmitted. The order of transmitted motion vector data is the same as that indicated in Figure2. For the code_number of motion vector data, please refer to Table 1.

7.5 Decoder Process for motion vector

7.5.1 Differential motion vectors

Motion vectors for forward, backward, or bi-directionally predicted macroblock are differentially encoded. A prediction has to be added to the motion vector differences to get the motion vectors for the macroblock. The predictions are formed in way similar to that described in section 3.6.2. The only difference is that forward motion vectors are predicted only from forward motion vectors in surrounding macroblocks, and backward motion vectors are predicted only from backward motion vectors in surrounding macroblocks.

If a neighboring macroblock does not have a motion vector of the same type or does not use the same reference frame for multiple reference frame prediction, the candidate predictor for that macroblock is set to zero for that motion vector type.

7.5.2 Motion vectors in direct mode

In direct mode the same block structure as for the macroblock in the temporally subsequent picture is assumed. For each of the subblocks the forward and backward motion vectors are computed as scaled versions of the corresponding vector components of the macroblock in the temporally subsequent picture as described below.

As the multiple reference frame prediction is used, the forward reference frame for the direct mode is the same as the one used for the corresponding macroblock in the temporally subsequent reference picture. The forward and backward motion vectors for direct mode macroblocks are calculated as follows.

$$MV_F = (TR_B * MV) / TR_D$$

$$MV_B = (TR_B - TR_D) * MV / TR_D$$

Where the vector component MV_F is the forward motion vectors, MV_B is the backward motion vector, and MV represents the motion vectors in the corresponding macroblock in the subsequent reference picture. Note that if the subsequent reference is an intra-coded frame or the reference macroblock is an intra-coded block, the motion vectors are set to zero. TR_D is the temporal distance between the temporally previous and next reference frame, and TR_B is the temporal distance between the current frame and previous reference frame.

It should be noted that when multiple reference frame prediction is used, the reference frame for the motion vector predictions is treated as though it were the most recent previous decoded frame. Thus, instead of using the temporal reference of the exact reference frame to compute the temporal distances TR_D and TR_B, the temporal reference in most recent previous reference frame is used to compute the temporal distances TR_D and TR_B.

8 Data Partitioning and Interim File Format

The traditional TML bit stream syntax simply concatenates the VLC coded symbols to form a bit stream, with the exception of the possible byte alignment of the picture (ans slice?) header. Data Partitioning re-arranges the symbols in such a way that all symbols of one data type (e.g. DC coefficients, macrobloc headers, motion vectors) that belong to a single slice are collected in one VLC coded bitstream that starts byte aligned. Decoders can process such a partitioned data streams by fetching symbols from the correct patition. The partition to fetch from is determined through the decoder's state machine, according to the syntax diagram discussed in section 2.4
In order to cleanly divide the concept of data partitioning (which is useful in all error prone environments) from network specifica, an interim file format is used that stores the compressed, VLC coded bits of each partition along with easy-to-process, uncompressed header information. It should be emphasized that this file format is not intended to be transmitted – it rather forms a clean interface between network specific and network unspecific syntax elements. Adaptation layers for IP and highly bit error prone H.223-based networks are discussed later in this document.

8.1 Data Partitioning

Data Partitioning is implemented by concatenating all VLC coded symbols of one data type and one slice (or full picture if slices are not used). At the moment, for a few partitions as indicated below contain data of more than one

data type that are so closely related that a finer diversion seems to be fruitless. The following data types are currently defined:

0	TYPE_HEADER	Picture or Slice Headers (Note 1)
1	TYPE_MBHEADER	Macroblock header information (Note 2)
2	TYPE_MVD	Motion Vector Data
3	TYPE_CBP	Coded Block Pattern
4	TYPE_2x2DC	2x2 DC Coefficients
5	TYPE_COEFF_Y	Luma AC Coefficients
6	TYPE_COEFF_C	Chroma AC Coefficients
7	TYPE_EOS	End-of-Stream Symbol

Note 1: TYPE_HEADER encompasses all Picture/Slice header information
Note 2: TYPE_MBHEADER encompasses The MB-Type, Intra-Prediction mode and Reference Frame ID.

8.2 Interim File Format

The Interim File Format consists of variable length records that include a fixed length header and the initially byte aligned, concatenated, VLC coded symbols. The fixed length header has the following structure:

int32	BitCount	Number of VLC coded bits following the fixed length header
int32	DataType	Data Type of symbols
int32	PicID	Picture ID, currently a copy of the TR
int32	SliceID	Slice ID, an integer value indicating that the informatuion belongs to the n'th slice of PicID
int32	StartMB	The starting MB (in scan order, starting at 0 ???) of that slice
int32	QP	The Quantizer Parameter (copy of the picture/slice header's QP, non-VLC coded
int32	Format	Picture Format, non-VLC coded copy of the picture header's Format field
[]byte	Data	Bitcount # of bits, padded with 0 bits in the last byte

The ordering of partitions is by picture, slice, data-type. If no symbols are coded for a given data type (which is frequently the case for Intra pictures) then that record is not present. Decoders should be designed to accept out-of-order data type records for a give Slice/Picture. Any other syntax violation in particular missing partitions, can be used to trigger syntax-based repair and/or error concealment.

9 Network Adaptation Layer for IP networks

This section covers the Network Adaptation Layer for non-managed, best effort IP networks using RTP [RFC1889] as the transport. The section will likely end up in the form of a standard's track RFC covering an RTP packetization for H.26L.

The NAL takes the information of the Interim File Format as discussed in section 8.2 and converts it into packets that can conveyed directly over RTP. It is designed to be able to take advantage of more than one virtual transport stream (either within one RTP stream by unequal packet content protection currently discussed in the IETF and as Annex I of H.323, or by using several RTP streams with network or application layer unequal error protection).

In doing so, it has to

- arrange partitions in an intelligent way into packets
- eventually split/recombine partitions to match MTU size constraints
- avoid the redundancy of (mandatory) RTP header information and information in the video stream
- define receiver/decoder reactions to packet losses. Note: the IETF tends more and more to do this in a normative way, whereas in the ITU and in MPEG this is typically left to implementers. Issue has to be discussed one day. Current document provides information without makeing any assumptions about that.

9.1 Assumptions

Any packetization scheme has to make some assumptions on typical network conditions and constraints. The following set of assumptions have been used in earlier Q.15 research on packetization and are deemed to be still valid:

- MTU size: around 1500 bytes per packet for anything but dial-up links, 500 bytes for dial-up links.

- Packet loss characteistic: non-bursty (due to drop-tail router implementations, and assuming reasonable pacing algorithms (e.g. no bursting occurs at the sender).
- Packet loss rate: up to 20%

9.2 Combining of Partitions according to Priorities

In order to allow uneqal protection of more important bits of the bitstream, exactly two packets per slice are generated (see Q15-J-53 for a detailed discussion). Slices should be used to ensure that both packets meet the MTU size constraints to avoid network splitting/recombining processes.

The 'First' packet contains the following partitions:
- TYPE_HEADER
- TYPE_MBHEADER
- TYPE_MVD
- TYPE_EOS

The 'Second' packet is assembled using the rest of the partitions
- TYPE_CBP Coded Block Patterm
- TYPE_2x2DC 2x2 DC Coefficients
- TYPE_COEFF_Y Luma AC Coefficients
- TYPE_COEFF_C Chroma AC Coefficients

This configuration allows decoding the first packet independently from the second (although not vice versa). As the first packet is more important both because motion information is important for struction [for what?] and because the 'First' packet is necessary to decode the 'Second', UEP should be used to protect the 'First' packet s higher.

9.3 Packet Structure

Each packet consists of a fixed 32 bit header a series of Part-of-Partition structures (POPs).

The packet header contains information

Bit 31	1 == This packet contains a picture header
Bit 30	1 == This packet contains a slice header
Bits 25..29	Reserved
Bits 10-24	StartMB. It is assumed that no picture has more than 2**14 Macroblocks
Bits 0..9	SliceID. It is assumed that a picture does not have more than 1024 slices

Note: The PictureID (TR) can be easily reconstructed from the RTP Timestamp and is therefore not coded again.
Note: The current software reconstructs QP and Format out of the VLC coded Picture/Slice header symbols. This is architecturally not nice and should be changed, probably by deleting these two values from the interim File Format.
Note: This header is likely to change once bigger picture formats etc. come into play.

Each Part-of-Partition structure contains a header of 16 bits whose format is as follows

Bits 15..12	Data Type
Bits 11..0	Length of VLC-coded POP payload (in bits, starting byte-aligned, 0 indicates 4096 bits of paylaod)

Th reasoning behind the introduction of POP packets lies in avoiding large fixed length headers for (typically) small partitions. See Q15-J-53.

9.4 Packetization Process

The packetization process converts the Interim File Format (or, in a real world system, data partitioned symbols arriving through a software interface) into packets. The following RTP header fields are used (see RFC1889 for exact semantics):
- Timestamp: is calculated according to the rules of RFC1889 and RFC2429 based on a 90 KHz timestamp.
- Marker Bit: set for the very last packet of a picture ('Second' packet of the last Slice), otherwise cleared.
- Sequence Number is increased by one for every generated packet, and starts with 0 for easier debugging (this in contrast to RFC1889, where a random initialization is mandatory for security purposes).
- Version (V): 2
- Padding (P): 0
- Extension (X): 0

- Csourcecount (CC): 0
- Payload Type (PT): 0 (This in contrast to RFC1889 where 0 is forbidden)

The RTP header is followed by the payload, which follows the packet structure of section 9.3.

The RTP packet file, used as the input to packet loss simulators and similar tools (note that this format is identical to the ones used for IP-related testing during the H.263++ project, so that the loss simulators, error paytterns etc, can be re-used):

- Int32 size of the following packet in bytes
- []byte packet content, starting with the RTP header.

9.5 De-packetization

The De-packetization process reconstructs a file in the Interim File Format from an RTP packet file (that is possible subject to packet losses). This task is straightforwad and reverse to the packetization process. (Note that the QP and Format fields curretly have to be reconstructed using the VLC-coded symbols in the TYPE_HEADER partition. This bug in the Interim File Format spec should be fixed some time).

9.6 Repair and Error Concealment

In order to take advantage of potential UEP for the 'First' packet and the ability of the decoder to reconstruct data where CBP/coefficient information was lost, a very simple error concealment strategy is used. This strategy repairs the bitstream by replacing a lost CBP partition with CBPs that indicate no coded coefficients. Unfortunately, the CBP codewords for Intra and Inter blocks are different, so that such a repair cannot be done context-insensitive. Instead of (partly) VLC-decoding the CBP partition in the NAL module in order to insert the correct type of CBP symbol in the (lost) partition, the decoder itself can be changed to report the appropriate CBP symbols saying "No coefficients" whenever the symbol fetch for a CBP symbol returns with the indication of a lost/empty partition.

INTERNATIONAL TELECOMMUNICATION UNION
TELECOMMUNICATION STANDARDIZATION SECTOR
STUDY PERIOD 1997 - 2000

COM 16-___-E
November, 2000
Original: English

Question: 15/16

STUDY GROUP 16 – CONTRIBUTION ___

SOURCE*: RAPPORTEUR FOR Q.15/16 (Gary SULLIVAN)

TITLE: DRAFT FOR "H.263++" ANNEXES U, V, AND W TO RECOMMENDATION H.263

This contribution contains a draft for three new annexes to be added to ITU-T Recommendation H.263 (*Video Coding for Low Bit Rate Communication*):

Annex U, specifying an optional *Enhanced Reference Picture Selection* (ERPS) mode,
Annex V, specifying an optional *Data Partitioned Slice* (DPS) mode, and
Annex W, specifying optional *Additional Supplemental Enhancement Information.*

Drafts of these annexes were determined by ITU-T SG16 in February, 2000. It is proposed that this draft of these annexes be decided at the SG 16 meeting in November 2000.

SUMMARY

This document contains three additional annexes to Recommendation H.263:

Annex U: An optional *Enhanced Reference Picture Selection* (ERPS) mode capable of providing enhanced coding efficiency and enhanced error resilience (particularly against loss of data packets). The ERPS mode operates by managing a multi-picture buffer of stored pictures.

Annex V: An optional *Data Partitioned Slice* (DPS) mode capable of providing enhanced error resilience (particularly against localized corruption of bitstream contents during transmission). The DPS mode operates by separating header and motion vector data from DCT coefficient data in the bitstream and by protecting motion vector data using a reversible representation.

Annex W: Optional *Additional Supplemental Enhancement Information* which can be added to an H.263 bitstream to provide backward-compatible enhancements, including:

- Indication of use of a specific fixed-point IDCT
- Picture Messages, including the message types of:
  - Arbitrary Binary Data,
  - Text (Arbitrary, Copyright, Caption, Video Description, or Uniform Resource Identifier),
  - Picture Header Repetition (Current, Previous, Next with Reliable Temporal Reference, or Next with Unreliable Temporal Reference),
  - Interlaced Field Indications (Top or Bottom), and
  - Spare Reference Picture Identification.

* **Contact:** Gary Sullivan
Microsoft Corporation
Redmond WA 98052
United States

Tel: +1 425 882 8080
Fax: +1 425 936 7329
Email: garysull@microsoft.com

Annex U
Enhanced Reference Picture Selection mode

(This annex forms an integral part of this Recommendation.)

U.1 Introduction

This annex describes the optional Enhanced Reference Picture Selection (ERPS) mode of this Recommendation. The capability to use this optional mode is negotiated by external means (for example, Recommendation H.245). The amount of picture memory accommodated in the decoder for ERPS operation should also be signaled by external means. The use of this mode shall be indicated by setting the formerly-reserved bit 16 of the optional part of the PLUSPTYPE (OPPTYPE) to "1". The mode provides benefits for both error resilience and coding efficiency by using a memory buffer of reference pictures.

A sub-mode of the ERPS mode is specified for Sub-Picture Removal. The purpose of Sub-Picture Removal is to reduce the amount of memory required to store multiple reference pictures. The memory reduction is accomplished by specifying the partitioning of each reference picture into smaller rectangular units called sub-pictures. The encoder can then indicate to the decoder that specific sub-picture areas of specific reference pictures will not be used as a reference for the prediction of subsequent pictures, thus allowing the memory allocated in the decoder for storing these areas to be used to store data from other reference pictures. The support for this sub-mode and the allowed fragmentation of the picture memory into minimum picture units (MPUs) for Sub-Picture Removal as defined herein is also negotiated by external means (for example, Recommendation H.245).

A sub-mode of the ERPS mode is specified for enabling two-picture backward prediction in B pictures. This sub-mode can enhance performance by providing encoders for B pictures not only with an ability to use multiple references for forward prediction, but also to use more than one reference picture for backward prediction. The support for this sub-mode is negotiated by external means (for example, Recommendation H.245).

For error resilience, the ERPS mode can use backward channel messages, which are signaled by external means (for example, Recommendation H.245) sent from a decoder to an encoder to inform the encoder which pictures or parts of pictures have been incorrectly decoded. The ERPS mode provides enhanced performance compared to the Reference Picture Selection (RPS) mode defined in Annex N. It shall not be used simultaneously with the RPS mode. (It can be used in such a way as to provide essentially the same functionality as the RPS mode.)

For coding efficiency, motion compensation can be extended to prediction from multiple pictures. The extension of motion compensation to multi-picture prediction is achieved by extending each motion vector by a picture reference parameter that is used to address a macroblock or block prediction region for motion compensation in any of the multiple reference pictures. The picture reference parameter is a variable length code specifying a relative buffer index. The reference pictures are assembled in a buffering scheme that is controlled by the encoder.

The ERPS mode shall not be used with the Syntax-based Arithmetic Coding mode (see Annex E) or the Data Partitioned Slice mode (see Annex V).

Once activated, the ERPS mode shall not be inactivated in subsequent pictures in the bitstream unless the initial inactivation occurs in an I or EI picture and any reactivation is also in an I or EI picture and is accompanied by a buffer reset (RESET equal to "1"). If inactivated, the entire contents of the ERPS multi-picture buffer shall be set to "unused" status.

U.2 Video source coding algorithm

The source coder of this mode is shown in generalized form in Figure U.1. This figure shows a structure that uses a number of picture memories.

The video source coding algorithm can be extended to multi-picture motion compensation. Enhanced coding efficiency may be achieved by allowing reference picture selection on the macroblock level. A picture buffering scheme with relative indexing is employed for efficient addressing of pictures in the multi-picture buffer. The multi-picture buffer control may work in two distinct types of operation.

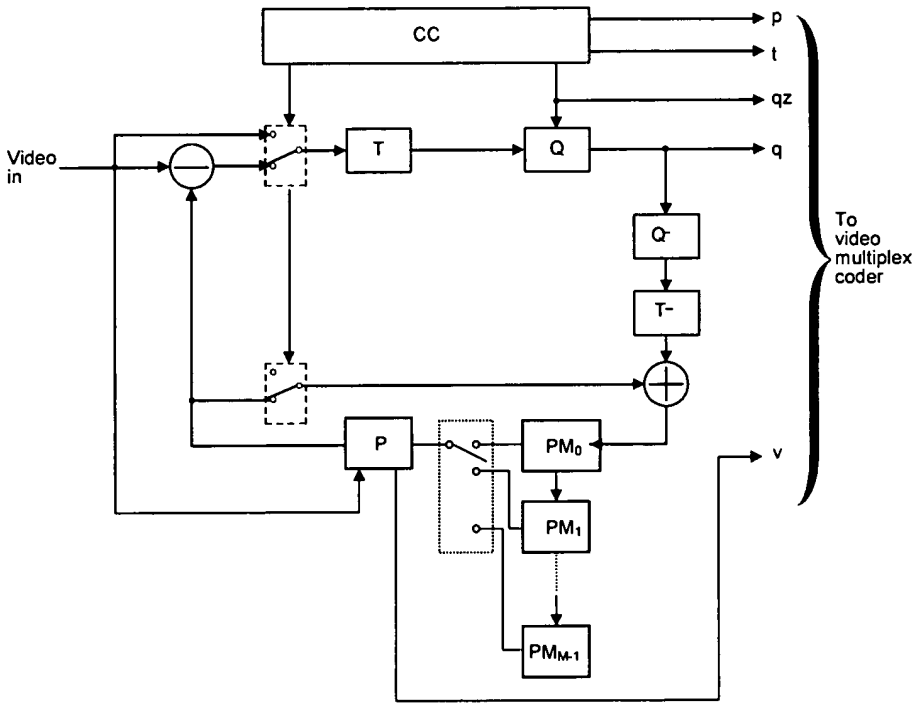
In the first of these two types of operation, a "Sliding Window" over time can be accommodated by the buffer control unit. In such a buffering scheme using M picture memories $PM_0 ... PM_{M-1}$, the most recent preceding (up to M) decoded and reconstructed pictures are stored in the picture memories and can be used as references for decoding. If the number of pictures maximally accommodated by the multi-picture buffer corresponds to M, the motion estimation when coding a picture m, if $0 \leq m \leq M-1$, can utilize m pictures. When coding a picture $m \geq M$, the maximum number of pictures M can be used. Alternatively, a

second "Adaptive Memory Control" type of operation can be used for a more flexible and specific control of the picture memories than with the simple "Sliding Window" scheme.

The operation of the ERPS mode results in the assignment of "unused" status to some pictures or sub-picture areas of pictures that have been sent to the decoder. Once some picture or area of a picture has been assigned to "unused" status, the bitstream shall not contain any data that causes a reference to any "unused" area for the prediction of subsequent pictures. By managing the assignment of "unused" status to previous pictures, the encoder shall ensure that sufficient memory is available in the decoder to store all data needed for the representation of subsequent pictures. The overall buffer size and structure is conveyed to the decoder in the bitstream, and the encoder shall control the buffer such that the specified total capacity is not exceeded by stored picture data that has not been assigned to "unused" status.

The source coder may select one or several of the picture memories to suppress temporal error propagation caused by inter-picture coding. The Independent Segment Decoding mode (see Annex R), which treats boundaries of GOBs with non-empty headers or slices as picture boundaries, can be used to avoid spatial error propagation due to motion compensation across the boundaries of the GOBs or slices when this mode is applied to a smaller unit than a picture, such as a GOB or slice. The information to signal which picture is selected for prediction is included in the encoded bitstream.

The strategy used by the encoder to select the picture or pictures to be used for prediction is out of the scope of this Recommendation.



T Transform
Q Quantizer
P Picture Memory with motion compensated variable delay
PM Picture Memory
CC Coding control
p Flag for INTRA/INTER
t Flag for transmitted or not
qz Quantizer indication
q Quantizing index for transform coefficients
v Motion vector

FIGURE U.1/H.263

Source coder for Enhanced Reference Picture Selection mode

U.3 Forward-Channel Syntax

The syntax is altered in the picture, Group of Blocks (GOB), and slice layers. When indicated by a parameter MRPA being equal to "1", the syntax is also altered in the macroblock layer. In the picture, GOB, and slice layers, an Enhanced Reference

Picture Selection layer (ERPS layer) is inserted. In the macroblock layer, picture reference parameters are inserted under certain conditions to enable multi-picture motion compensation.

U.3.1 Syntax of the Picture, GOB, and Slice layer

The Enhanced Reference Picture Selection syntax for the PLUS header (otherwise as shown in Figure 8) is shown in Figure U.2. The fields of RPSMF, PN, and the ERPS layer are inserted into the PLUS header. The fields of TRPI, TRP, BCI, and BCM are not present (since they are only needed for the RPS mode of Annex N, which is not allowed when the ERPS mode is active).

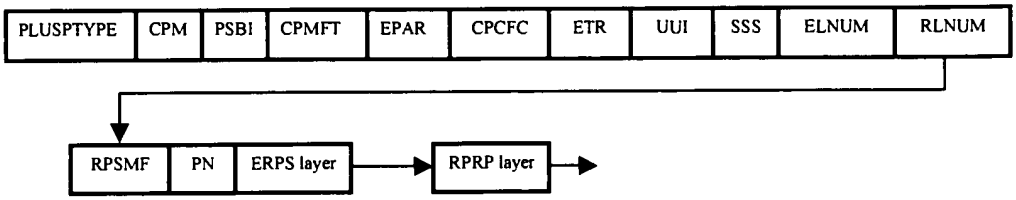


FIGURE U.2/H.263

Structure of PLUS Header for the ERPS mode

The syntax for the GOB layer is shown in Figure U.3. The fields of PNI, PN, NOERPSL, and the ERPS layer are added to the syntax (otherwise defined as in Figure 9).

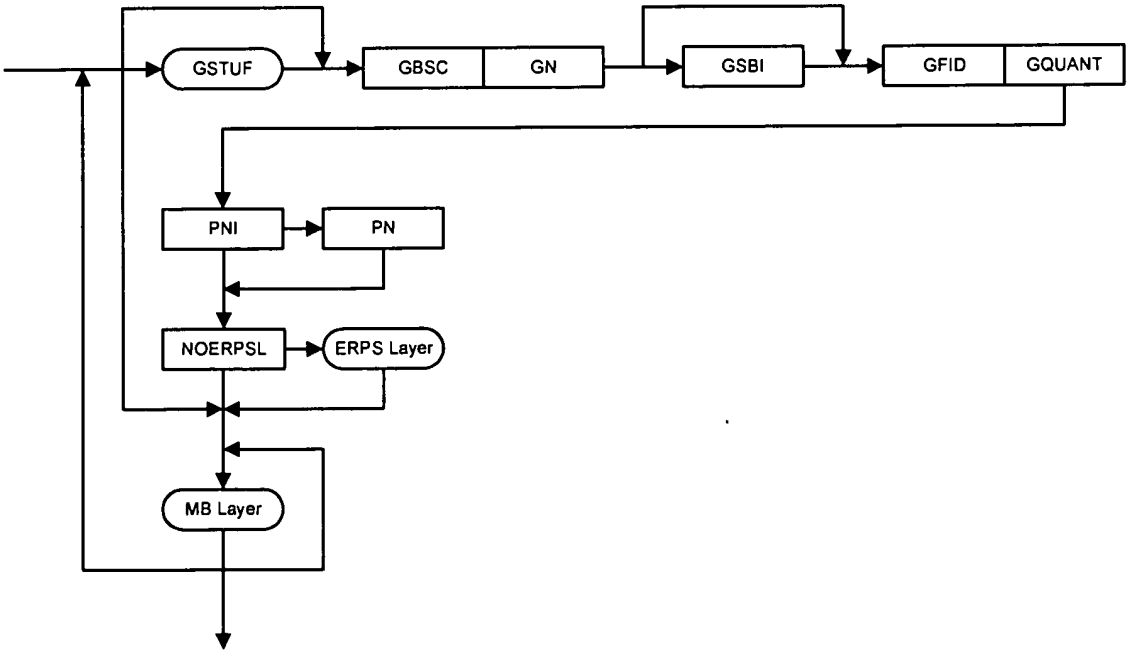


FIGURE U.3/H.263

Structure of GOB layer for the ERPS mode

When the optional Slice Structured mode (see Annex K) is in use, the syntax of the slice layer is modified in the same way as the GOB layer. The syntax for the slice layer is shown in Figure U.4. The slice that immediately follows the picture start code in the bitstream also includes all of the added fields PNI, PN, NOERPSL, and the ERPS layer.

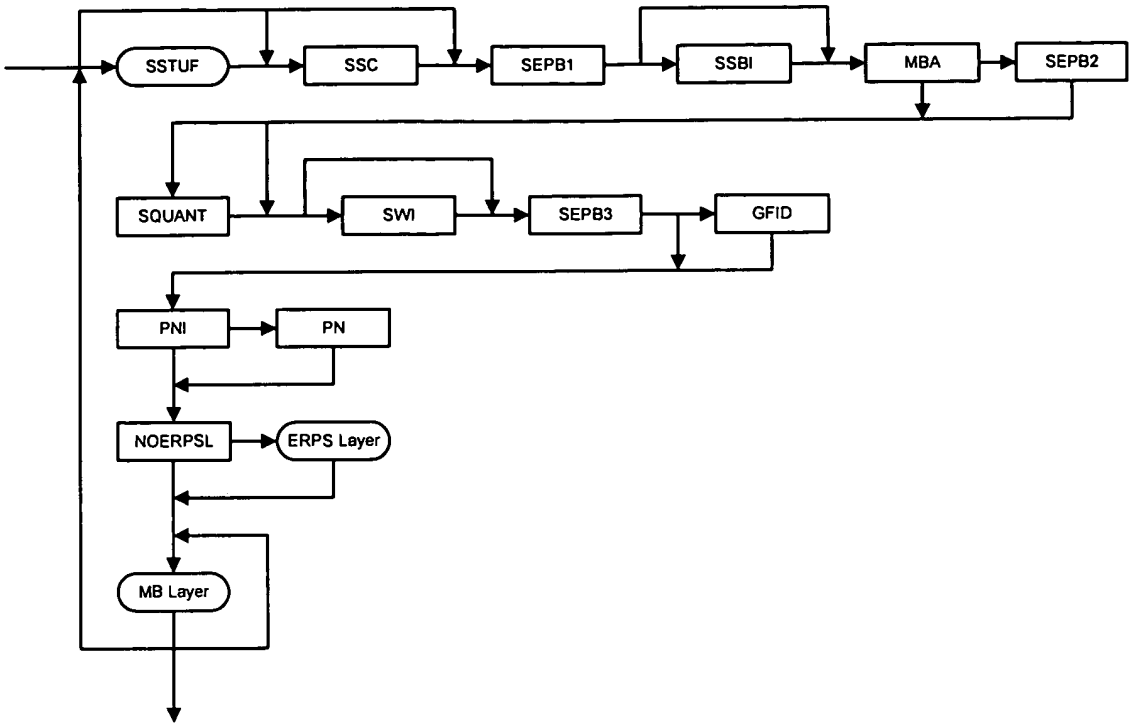


FIGURE U.4/H.263

Structure of Slice layer for the ERPS mode

The ERPS layer is shown in Figure U.5.

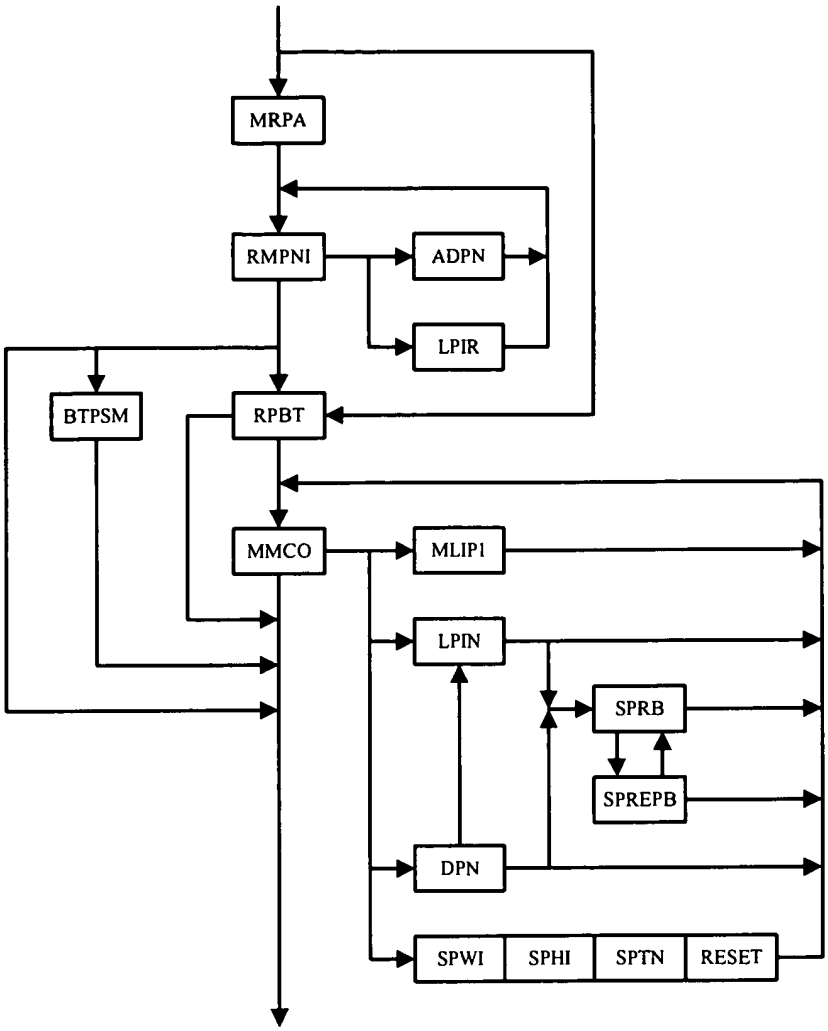


FIGURE U.5/H.263

Structure of the ERPS layer

Variable length codes for the ADPN, LPIR, MLIP1, DPN, LPIN, SPTN, PR, PR_0, PR_2, PR_3, PR_4, PRB, and PRFW fields are given in Table U.1.

Table U.1/H.263

Variable length codes for ADPN, LPIR, MLIP1, DPN, LPIN, SPTN, PR, PR_0, PR_2, PR_3, PR_4, PRB, and PRFW

Absolute position	Number of bits	Codes
0	1	1
"x_0"+1 (1:2)	3	$0x_00$
"x_1x_0"+3 (3:6)	5	$0x_11x_00$
"$x_2x_1x_0$"+7 (7:14)	7	$0x_21x_11x_00$
"$x_3x_2x_1x_0$"+15 (15:30)	9	$0x_31x_21x_11x_00$
"$x_4x_3x_2x_1x_0$"+31 (31:62)	11	$0x_41x_31x_21x_11x_00$
"$x_5x_4x_3x_2x_1x_0$"+63 (63:126)	13	$0x_51x_41x_31x_21x_11x_00$
"$x_6x_5x_4x_3x_2x_1x_0$"+127 (127:254)	15	$0x_61x_51x_41x_31x_21x_11x_00$
"$x_7x_6x_5x_4x_3x_2x_1x_0$"+255 (255:510)	17	$0x_71x_61x_51x_41x_31x_21x_11x_00$
"$x_8x_7x_6x_5x_4x_3x_2x_1x_0$"+511 (511:1022)	19	$0x_81x_71x_61x_51x_41x_31x_21x_11x_00$
"$x_9x_8x_7x_6x_5x_4x_3x_2x_1x_0$"+1023 (1023:2046)	21	$0x_91x_81x_71x_61x_51x_41x_31x_21x_11x_00$
"$x_{10}x_9x_8x_7x_6x_5x_4x_3x_2x_1x_0$"+2047 (2047:4094)	23	$0x_{10}1x_91x_81x_71x_61x_51x_41x_31x_21x_11x_00$

U.3.1.1 Reference Picture Selection Mode Flags (RPSMF) (3 bits)
RPSMF is a 3 bit fixed length codeword that is present in the PLUS header whenever the ERPS mode is in use (regardless of the value of UFEP). RPSMF shall not be present in the GOB or slice layer. When present, RPSMF indicates which type of back-channel messages are needed by the encoder. The values of RPSMF shall be as defined in subclause 5.1.13.

U.3.1.2 Picture Number Indicator (PNI) (1 bit)
PNI is a single bit fixed length codeword that is always present at the GOB or slice layer and is not present in the PLUS header. When present, PNI indicates whether or not the following PN field is also present.

"0": PN field is not present.
"1": PN field is present.

U.3.1.3 Picture Number (PN) (10 bits)
PN is a 10 bit fixed length codeword that is always present in the PLUS header when the ERPS mode is in use, and is present at the GOB or slice layer only when indicated by PNI.

PN shall be incremented by 1 for each coded and transmitted picture, in a 10-bit modulo operation, relative to the PN of the previous stored picture. The term "stored picture" is defined in subclause U.3.1.5.7. For EI and EP pictures, PN shall be incremented from the value in the last stored EI or EP picture within the same scalability enhancement layer. For B pictures, PN shall be incremented from the value in the most temporally-recent stored non-B picture in the reference layer of the B picture which precedes the B picture in bitstream order (a picture which is temporally subsequent to the B picture). B pictures are not stored in the multi-picture buffer, as they are not used as references for subsequent pictures. Thus a picture immediately following a B picture in the reference layer of the B picture or another B picture which immediately follows a B picture in the same enhancement layer shall have the same PN as the B picture. Similarly, if a non-B picture is present in the bitstream which is not stored, the picture following this non-B picture (in the same enhancement layer, in the case of Annex O operation) shall have the same PN as the non-stored non-B picture.

In a usage scenario known as "Video Redundancy Coding", the ERPS mode may be used by some encoders in a manner in which more than one representation is sent for the pictured scene at the same temporal instant (usually using different reference pictures). In such a case in which the ERPS mode is in use and in which adjacent pictures in the bitstream have the same temporal reference and the same picture number, the decoder shall regard this occurrence as an indication that redundant copies have been sent of approximately the same pictured scene content, and shall decode and use the first such received picture while discarding the subsequent redundant picture(s).

The PN serves as a unique ID for each picture stored in the multi-picture buffer (for a given enhancement layer, in the case of Annex O operation) within 1024 coded and stored pictures. Therefore, a picture cannot be kept in the buffer after more than 1023 subsequent coded and stored pictures (in the same enhancement layer, in the case of Annex O operation) unless it has

been assigned a long-term picture index as specified below. The encoder shall ensure that the bitstream shall not specify retaining any short-term picture after more than 1023 subsequent stored pictures. A decoder which encounters a picture number on a current picture having a value equal to the picture number of some other short-term stored picture in the multi-picture buffer (in the same enhancement layer, in the case of Annex O operation, and excluding the video redundancy coding case described in the previous paragraph) should treat this condition as an error.

U.3.1.4 No Enhanced Reference Picture Selection Layer (NOERPSL) (1 bit)

NOERPSL is a single bit fixed length codeword that is present at the GOB or slice level whenever the ERPS mode is in use. It is not present in the PLUS header. The values of NOERPSL shall be as follows:

"0": The ERPS layer is sent,
"1": The ERPS layer is not sent.

If NOERPSL is "1", all ERPS settings and re-mappings in effect for the picture shall be applied also for the relevant GOB or slice. ERPS layer information sent at the GOB or slice level does not affect the decoding process of any other GOB or slice.

U.3.1.5 Enhanced Reference Picture Selection layer (ERPS) (variable length)

The ERPS layer is always present at the picture level when the ERPS mode is in use, and is present at the GOB or slice level if NOERPSL is "0". It specifies the buffer indexing used to decode the current picture, GOB, or slice, and manages the contents of the picture buffer.

U.3.1.5.1 Multiple Reference Pictures Active (MRPA) (1 bit)

MRPA is a single bit fixed length codeword that is present only if the picture coding type indicates a P-picture, an EP-picture, an Improved PB frame, or B-picture. MRPA is the first element in the ERPS layer if present. MRPA specifies whether the number of active reference pictures for forward-prediction or backward-prediction decoding of the current picture, GOB, or slice may be larger than one. The value of MRPA shall be as follows:

"1": More than one reference picture may be used for forward or backward motion compensation.
"0": Only one reference picture is used for forward or backward motion compensation. In this case, the extensions of the macroblock layer syntax in subclause U.3.2 do not apply.

MRPA may be changed from GOB to GOB or slice to slice so that different GOBs or slices may address different numbers of reference pictures.

MRPA shall be "0" in any picture which invokes the Reference Picture Resampling mode (see Annex P), and the same picture shall be indicated as the forward reference picture to be used at both the picture and GOB or slice levels for any such current picture. If the current picture is a B picture, the backward reference picture shall have the same size as the current picture, and any reference picture resampling process shall be applied only to the forward reference picture. Reference picture resampling shall be invoked only if the multi-picture buffer contains sufficient "unused" capacity to store the resampled forward reference picture, but after the resampled reference picture is used for the decoding of the current picture, the resampled forward reference picture shall not be stored in the multi-picture buffer.

U.3.1.5.2 Re-Mapping of Picture Numbers Indicator (RMPNI) (variable length)

RMPNI is a variable length codeword that is present in the ERPS layer if the picture is a P, EP, Improved PB, or B picture. RMPNI indicates whether any default picture indices are to be re-mapped for motion compensation of the current picture, GOB, or slice – and how the re-mapping of the relative indices into the multi-picture buffer is to be specified if indicated. RMPNI is transmitted using Table U.2. If RMPNI indicates the presence of an ADPN or LPIR field, an additional RMPNI field immediately follows the ADPN or LPIR field.

A picture reference parameter is a relative index into the ordered set of pictures. The RMPNI, ADPN, and LPIR fields allow the order of that relative indexing into the multi-picture buffer to be temporarily altered from the default index order for the decoding of a particular picture, GOB, or slice. The default index order is for the short-term pictures (i.e., pictures which have not been given a long-term index) to precede the long-term pictures in the reference indexing order. Within the set of short-term pictures, the default order is for the pictures to be ordered starting with the most recent buffered reference picture and proceeding through to the oldest reference picture (i.e., in decreasing order of picture number in the absence of wrapping of the ten-bit picture number field). Within the set of long-term pictures, the default order is for the pictures to be ordered starting with the picture with the smallest long-term index and proceeding up to the picture with long-term index equal to the most recent value of MLIP1–1.

For example, if the buffer contains three short-term pictures with short-term picture numbers 300, 302, and 303 (which were transmitted in increasing picture-number order) and two long-term pictures with long-term picture indices 0 and 3, the default index order is:

- default relative index 0 refers to the short-term picture with picture number 303,
- default relative index 1 refers to the short-term picture with picture number 302,
- default relative index 2 refers to the short-term picture with picture number 300,
- default relative index 3 refers to the long-term picture with long-term picture index 0, and
- default relative index 4 refers to the long-term picture with long-term picture index 3.

The first ADPN or LPIR field that is received (if any) moves a specified picture out of the default order to the relative index of zero. The second such field moves a specified picture to the relative index of one, etc. The set of remaining pictures not moved to the front of the relative indexing order in this manner shall retain their default order amongst themselves and shall follow the pictures that have been moved to the front of the buffer in relative indexing order.

If MRPA is "0", no more than one ADPN or LPIR field shall be present in the same ERPS layer unless the current picture is a B picture. If the current picture is a B picture and MRPA is "0", no more than two ADPN or LPIR fields shall be present in the same ERPS layer.

Any re-mapping of picture numbers specified for some picture shall not affect the decoding process for any other picture. Any re-mapping of picture numbers specified for some GOB or slice shall not affect the decoding process for any other GOB or slice. A re-mapping of picture numbers specified for a picture shall only affect the decoding process for any GOB or slice within that picture in two ways:

- If NOERPSL is "1" at the GOB or slice level, then the re-mapping specified at the picture level is also used at the GOB or slice level.
- If the picture is a B picture, the re-mapping specified at the picture level shall specify the calculation of the value of TR_B and TR_D for direct bidirectional prediction.

An RMPNI "end loop" indication is the last element of the ERPS layer for a B picture if MRPA is "0". In a B picture with MRPA equal to "1", an RMPNI "end loop" indication is followed by BTPSM. In a P or EP picture or Improved PB frame, an RMPNI "end loop" indication is followed by RPBT.

Within one ERPS layer, RMPNI shall not specify the placement of any individual reference picture into more than one re-mapped position in relative index order.

Table U.2/H.263

RMPNI operations for re-mapping of reference pictures

Value	Re-mapping Specified
'1'	ADPN field is present and corresponds to a negative difference to add to a picture number prediction value
'010'	ADPN field is present and corresponds to a positive difference to add to a picture number prediction value
'011'	LPIR field is present and specifies the long-term index for a reference picture
'001'	End loop for re-mapping of picture relative indexing default order

U.3.1.5.3 Absolute Difference of Picture Numbers (ADPN) (variable length)

ADPN is a variable length codeword that is present only if indicated by RMPNI. ADPN follows RMPNI when present. ADPN is transmitted using Table U.1, where the index into the table corresponds to ADPN – 1. ADPN represents the absolute difference between the picture number of the currently re-mapped picture and the prediction value for that picture number. If no previous ADPN fields have been sent within the current ERPS layer, the prediction value shall be the picture number of the current picture. If some previous ADPN field has been sent, the prediction value shall be the picture number of the last picture that was re-mapped using ADPN.

If the picture number prediction is denoted PNP, and the picture number in question is denoted PNQ, the decoder shall determine PNQ from PNP and ADPN in a manner mathematically equivalent to the following:

```
if (RMPNI == '1') {// a negative difference
  if (PNP – ADPN < 0)
```

```
  PNQ = PNP – ADPN +1024;
 else
  PNQ = PNP – ADPN;
}else{              // a positive difference
 if (PNP + ADPN > 1023)
  PNQ  = PNP + ADPN – 1024;
 else
  PNQ  = PNP + ADPN;
}
```

The encoder shall control RMPNI and ADPN such that the decoded value of ADPN shall not be greater than or equal to 1024.

As an example implementation, the encoder may use the following process to determine values of ADPN and RMPNI to specify a re-mapped picture number in question, PNQ:

```
DELTA = PNQ – PNP;
if (DELTA < 0) {
 if (DELTA < –511)
  MDELTA = DELTA + 1024;
 else
  MDELTA = DELTA;
}else{
 if(DELTA > 512)
  MDELTA = DELTA – 1024;
 else
  MDELTA = DELTA;
}

ADPN = abs(MDELTA);
```

where abs() indicates an absolute value operation. Note that the index into Table U.1 corresponds to the value of ADPN – 1, rather than the value of ADPN itself.

RMPNI would then be determined by the sign of MDELTA.

U.3.1.5.4 Long-term Picture Index for Re-Mapping (LPIR) (variable length)

LPIR is a variable length codeword that is present only if indicated by RMPNI. LPIR follows RMPNI when present. LPIR is transmitted using Table U.1. It represents the the long-term picture index to be re-mapped. The prediction value used by any subsequent ADPN re-mappings is not affected by LPIR.

U.3.1.5.5 B-Picture Two-Picture Prediction Sub-Mode (BTPSM) (1 bit)

BTPSM is a single bit fixed length codeword that is present only in a B picture (see Annex O) and only when MRPA is "1". It follows an RMPNI "end loop" indication and is the last element of the ERPS layer for the B picture when present. It indicates whether the two-picture backward prediction sub-mode is in use for the picture as follows:

"0" : Single-picture backward prediction
"1" : Two-picture backward prediction

BTPSM has an implied value of "0" if not present (when MRPA is "0").

The set of pictures available for use as forward prediction references is the set of pictures in the multi-picture buffer other than the set of backward reference pictures. The set of backward reference pictures is determined by the value of BTPSM. If single-picture backward prediction is specified by BTPSM, the first picture in (possibly re-mapped) relative index order is the only backward reference picture. If two-picture backward prediction is specified by BTPSM, the first two pictures in (possibly re-mapped) relative index order are the two backward reference pictures. The relative index for forward prediction then becomes a relative index into the set of forward reference pictures.

The contents of the multi-picture buffer are not affected by the presence of a B picture. The B picture is not stored in the multi-picture buffer and is not used as a reference for the coding of subsequent pictures.

U.3.1.5.6 Reference Picture Buffering Type (RPBT) (1 bit)

RPBT is a single bit fixed length codeword that specifies the buffering type of the currently decoded picture. It follows an RMPNI "end loop" indication when the picture is not an I, EI, or B picture. It is the first element of the ERPS layer if the picture is an I or EI picture. It is not present if the picture is a B picture. The values for RPBT are defined as follows:

"1" : Sliding Window,
"0" : Adaptive Memory Control.

In the "Sliding Window" buffering type, the current decoded picture shall be added to the buffer with default relative index 0, and any marking of pictures as "unused" in the buffer is performed automatically in a first-in-first-out fashion among the set of short-term pictures. In this case, if the buffer has sufficient "unused" capacity to store the current picture, no additional pictures shall be marked as "unused" in the buffer. If the buffer does not have sufficient "unused" capacity to store the current picture, the picture (or pictures as necessary to free the needed amount of memory in the case of sub-picture removal) with the largest default relative index (or indices as necessary in the case of sub-picture removal) among the short-term pictures in the buffer shall be marked as "unused". In the sliding window buffering type, no additional information is transmitted to control the buffer contents.

In the "Adaptive Memory Control" buffering type, the encoder explicitly specifies any addition to the buffer or marking of data as "unused" in the buffer, and may also assign long-term indices to short-term pictures. The current picture and other pictures may be explicitly marked as "unused" in the buffer, as specified by the encoder. This buffering type requires further information that is controlled by memory management control operation (MMCO) parameters.

RPBT, if present in GOB or slice layers, shall be the same as in the picture layer. Any MMCO command present in GOB or slice layers shall convey the same operation as some MMCO command in the picture layer.

If the picture is a B picture, RPBT shall not be present and the decoded picture shall not be stored in the multi-picture buffer. This ensures that a B picture shall not affect the contents of the multi-picture buffer.

Similarly, the B-picture part of an Improved PB frame shall not be stored in the buffer. All control fields associated with controlling the storage of an Improved PB frame shall be considered to be associated with controlling the storage of only the P-picture part of the Improved PB frame.

U.3.1.5.7 Memory Management Control Operation (MMCO) (variable length)

MMCO is a variable length codeword that is present only when RPBT indicates "Adaptive Memory Control", and may occur multiple times if present. It specifies a control operation to be applied to manage the multi-picture buffer memory. The MMCO parameter is followed by data necessary for the operation specified by the value of MMCO, and then an additional MMCO parameter follows – until the MMCO value indicates the end of the list of such operations. MMCO commands do not affect the buffer contents or the decoding process for the decoding of the current picture – rather, they specify the necessary buffer status for the decoding of subsequent pictures in the bitstream. The values and control operations associated with MMCO are defined in Table U.3.

All memory management control operations specified using MMCO shall be specified in the picture layer. Some or all of the same operations as are specified at the picture layer may also be specified at the GOB or slice layer (with the same associated data). MMCO shall not specify memory operations at the GOB or slice layer that are not also specified with the same associated data at the picture layer.

A buffer size and structure specification MMCO command shall be the first MMCO command if present. No more than one buffer size and structure specification MMCO command shall be present in a given ERPS layer. A buffer size and structure specification MMCO command with RESET equal to "1" shall be present in the first picture in which the ERPS mode is activated in any series of ERPS mode pictures in the bitstream. A buffer size and structure specification MMCO command with RESET equal to "1" shall precede any use of MMCO to indicate marking sub-picture areas of any short-term or long-term pictures as "unused". The sub-picture width and height specified in a buffer size and structure specification MMCO command shall not differ from the value of these parameters in a prior buffer size and structure specification MMCO command unless the current picture is an I or EI picture with RESET equal to "1". The picture height and width shall not change within the bitstream except within a picture containing a buffer size and structure specification MMCO command with RESET equal to "1" (or within a picture in which the ERPS mode is not in use).

If a B picture using single-picture backward prediction is present in the bitstream, exactly one temporally subsequent non-B picture in the reference layer of the B picture shall precede the B picture in bitstream order, as specified in subclause O.2. No memory management control operations shall be present within any ERPS layer of this immediately temporally subsequent non-B picture within the reference layer of the B picture which mark any part of that immediately temporally succeeding non-B picture as "unused", since that reference layer picture is needed for display until after the decoding of the B picture.

The transmission order constraints specified in subclause O.2 are adjusted as necessary for B pictures using two-picture backward prediction. If a B picture using two-picture backward prediction is present in the bitstream, exactly two temporally

subsequent non-B pictures in the reference layer of the B picture shall precede the B picture in bitstream order. The other restrictions on the transmission order of the B picture in the bitstream specified in O.2 shall apply, but as adjusted for the use of two temporally subsequent reference layer pictures. No memory management control operations shall be present within any ERPS layer of these two immediately temporally subsequent two non-B pictures within the reference layer of the B picture which mark any part of these two non-B pictures as "unused", since these reference layer pictures are needed for display until after the decoding of the B picture.

A "stored picture" is defined as a non-B picture which does not contain an MMCO command in its ERPS layer which marks that (entire) picture as "unused". If the current picture is not a stored picture, its ERPS layer shall not contain any of the following types of MMCO commands:

- An MMCO command to specify the buffer size and structure with RESET equal to "1",
- Any MMCO command which marks any other picture (other than the current picture) as "unused" that has not also been marked as "unused" in the ERPS layer of a prior stored picture,
- Any MMCO command which assigns a long-term index to a picture that has not also been assigned the same long-term index in the ERPS layer of a prior stored picture, or
- Any MMCO command which marks sub-picture areas of any picture as "unused" that have not also been marked as "unused" in the ERPS layer of a prior stored picture.

Table U.3/H.263

Memory Management Control Operation (MMCO) Values

Value	Memory Management Control Operation	Associated Data Fields Following
'1'	End MMCO Loop	None (end of ERPS layer)
'011'	Mark a Short-Term Picture as "Unused"	DPN
'0100'	Mark a Long-Term Picture as "Unused"	LPIN
'0101'	Assign a Long-Term Index to a Picture	DPN and LPIN
'00100'	Mark Short-Term Sub-Picture Areas as "Unused"	DPN and SPRB
'00101'	Mark Long-Term Sub-Picture Areas As "Unused"	LPIN and SPRB
'00110'	Specify the Maximum Long-Term Picture Index	MLIP1
'00111'	Specify the Buffer Size and Structure	SPWI, SPHI, SPTN, and RESET

U.3.1.5.8 Difference of Picture Numbers (DPN) (variable length)

DPN is present when indicated by MMCO. DPN follows MMCO if present. DPN is transmitted using codewords in Table U.1 and is used to calculate the PN of a picture for a memory control operation. It is used in order to assign a long-term index to a picture, mark a short-term picture as "unused", or mark sub-picture areas of a short-term picture as "unused". If the current decoded picture number is PNC and the decoded value from Table U.1 is DPN, an operation mathematically equivalent to the following equations shall be used for calculation of PNQ, the specified picture number in question:

```
if (PNC – DPN < 0)
  PNQ = PNC – DPN + 1024;
else
  PNQ = PNC – DPN;
```

Similarly, the encoder may compute the DPN value to encode using the following relation:

```
if (PNC – PNQ < 0)
  DPN = PNC – PNQ + 1024;
else
  DPN = PNC – PNQ;
```

For example, if the decoded value of DPN is zero and MMCO indicates marking a short-term picture as "unused", the current decoded picture shall be marked as "unused" (thus indicating that the current picture is not a stored picture).

U.3.1.5.9 Long-term Picture Index (LPIN) (variable length)

LPIN is present when indicated by MMCO. LPIN is transmitted using codewords in Table U.1 and specifies the long-term picture index of a picture. It follows DPN if the operation is to assign a long-term index to a picture. It follows MMCO if the operation is to mark a long-term picture as "unused" or to mark sub-picture areas of a long-term picture as "unused".

U.3.1.5.10 Sub-Picture Removal Bit-Map (SPRB) (fixed length)
SPRB is a fixed length codeword that contains one bit for each sub-picture area of a picture and is present when indicated by MMCO. The number of bits of SPRB data is determined by the most recent values of SPWI and SPHI. SPRB is used to indicate which sub-picture areas of a buffered picture are to be marked as "unused". SPRB follows DPN if the operation is to mark sub-picture areas of a short-term picture as "unused", and follows LPIN if the operation is to mark sub-picture areas of a long-term picture as "unused".

Sub-pictures are numbered in raster scan order starting from the upper-left corner of the picture. For example, consider a case in which a reference picture, specified by DPN, is partitioned into six sub-pictures. Let "s_1 s_2 s_3 s_4 s_5 s_6" represent six bits of SPRB data. If bit s_i is "1", then the decoder should mark the ith sub-picture in the indicated reference picture as "unused". For example, if the SPRB is '000110', then the fourth and fifth sub-pictures areas are marked as "unused".

To prevent start code emulation, all necessary SPREPB emulation prevention bits shall be inserted within or following the SPRB data as specified in subclause U.3.1.5.11.

If SPRB is present and the specified picture has previously been affected by a prior SPRB bit-map, the bit-map specified by SPRB shall contain a "1" for any sub-picture area that contained a "1" in that previous SPRB bit-map. Every SPRB bit-map shall contain at least one bit having the value "0" and at least one bit having the value "1".

U.3.1.5.11 Sub-Picture Removal Emulation Prevention Bit (SPREPB) (one bit)
SPREPB is a single bit fixed length codeword having the value "1" which shall be inserted immediately after any string of 8 consecutive zero bits of SPRB data.

U.3.1.5.12 Maximum Long-Term Picture Index Plus 1 (MLIP1) (variable length)
MLIP1 is a variable length codeword that is present if indicated by MMCO. MLIP1 follows MMCO if present. MLIP1 is transmitted using codewords in Table U.1. If present, MLIP1 is used to determine the maximum index allowed for long-term reference pictures (until receipt of another value of MLIP1). The decoder shall initially assume MLIP1 is "0" until some other value has been received. Upon receiving an MLIP1 parameter, the decoder shall consider all long-term pictures having indices greater than the decoded value of MLIP1 – 1 as "unused" for referencing by the decoding process for subsequent pictures. For all other pictures in the multi-picture buffer, no change of status shall be indicated by MLIP1.

U.3.1.5.13 Sub-Picture Width Indication (SPWI) (7 bits)
SPWI is a fixed length codeword of 7 bits that is present if indicated by MMCO. SPWI follows MMCO when indicated. SPWI specifies the width of a sub-picture in units of 16 luminance samples, such that the indicated sub-picture width is 16·(SPWI+1) luminance samples. The current picture has a width in sub-picture units of ceil(ceil(pw / 16) / (SPWI+1)) sub-pictures, where pw is the width of the picture and "/" indicates floating-point division. For positive numbers, the ceiling function, ceil(x), equals x if x is an integer and otherwise ceil(x) equals one plus the integer part of x. If a minimum picture unit (MPU) size defining the minimum width and height of a sub-picture has been negotiated by external means (for example, Recommendation H.245), the sub-picture width specified by SPWI shall be an integer multiple of the width of the MPU; otherwise, the sub-picture width specified by SPWI shall be such that SPWI is equal to ceil(pw / 16) – 1.

U.3.1.5.14 Sub-Picture Height Indication (SPHI) (7 bits)
SPHI is a fixed length codeword of 7 bits that is present if SPWI is present (as indicated by MMCO). SPHI follows SPWI if present. SPWI specifies the height of a sub-picture in units of 16 luminance samples, such that the indicated sub-picture height is 16·SPHI. The allowed range of values of SPHI is from 1 to 72. The current picture has a height of ceil(ceil(ph / 16) / SPHI) sub-pictures, where ph is the height of the picture and "/" indicates floating-point division. If a minimum picture unit (MPU) size defining the minimum width and height of a sub-picture has been negotiated by external means (for example, Recommendation H.245), the sub-picture height specified by SPHI shall be an integer multiple of the height of the MPU; otherwise, the sub-picture height specified by SPHI shall be such that SPHI is equal to ceil(ph / 16).

U.3.1.5.15 Sub-Picture Total Number (SPTN) (variable length)
SPTN is a variable length codeword that is present if SPWI and SPHI are present (as indicated by MMCO). SPTN follows SPHI if present. SPTN is coded using Table U.1, where the index into Table U.1 corresponds to the decoded value of SPTN – 1. The decoded value of STPN is the total operational size capacity of the multi-picture buffer in units of sub-pictures as specified by SPWI and SPHI. The memory capacity needed for the decoding of current pictures is not included in SPTN – only the memory capacity needed for storing the reference pictures to use for the prediction of other pictures. When sub-picture removal is not in use (i.e. when SPWI and SPHI have whole-picture dimensions), the maximum number of active short-

term reference pictures (for example, for sliding window operation) is thus given by SPTN minus the number of pictures that have been assigned to long-term indices and have not been subsequently marked as "unused".

U.3.1.5.16 Buffer Reset Indicator (RESET) (1 bit)

RESET is a single bit fixed length codeword that is present if SPWI, SPHI, and SPTN are present (as indicated by MMCO). RESET follows SPTN if present. The values of RESET shall be as follows:

"0": The buffer contents are not reset,
"1": The buffer contents are reset.

If RESET is "1", all pictures in the multi-picture buffer (but not the current picture unless specified separately) shall be marked "unused" (including both short-term and long-term pictures).

U.3.2 Macroblock layer syntax

U.3.2.1 P-Picture and Improved PB frames Macroblock Syntax

The macroblock layer syntax is modified if the ERPS layer is present for P-pictures and Improved PB frames when the number of selected forward reference pictures may be greater than one, as indicated by MRPA. The field MRPA is signaled in the ERPS layer. The macroblock layer syntax is shown in Figure U.7 when MRPA is "1". Otherwise, the macroblock syntax format in a P picture or Improved PB frame is not altered from that shown in Figure 10.

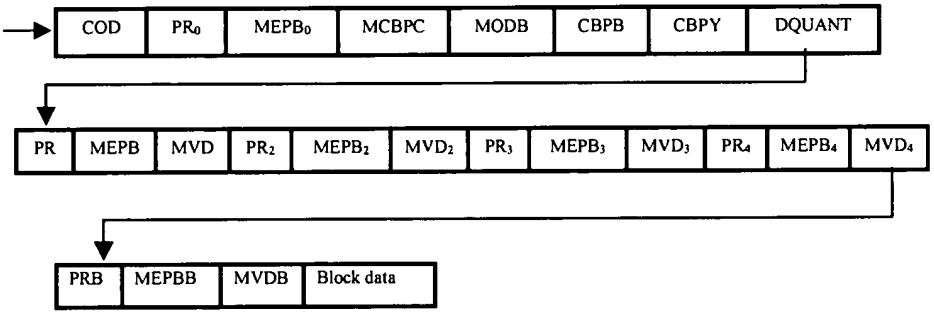


FIGURE U.7/H.263

Structure of P-picture and Improved PB frame Macroblock layer for the ERPS mode

U.3.2.1.1 Interpretation of COD

If the COD bit is "1", no further information is transmitted for the macroblock. In that case, the decoder shall treat the macroblock as an INTER macroblock with the motion vector for the entire macroblock equal to zero, picture reference parameter equal to zero, and with no coefficient data. If the COD bit is "0", indicating that the macroblock is coded, the syntax of the macroblock layer is depicted in Figure U.7 with the fields PR_0, PR, PR_2, PR_3, PR_4, and PRB being included in the syntax. PR_0, PR, PR_2, PR_3, PR_4, and PRB each consist of a variable length codeword as given in Table U.1.

U.3.2.1.2 Picture Reference Parameter 0 (PR_0) (variable length)

PR_0 is a variable length codeword as specified in Table U.1. It is present whenever COD is "0". If PR_0 has a decoded value of zero (codeword "1"), it indicates that further information will follow for the macroblock. If decoded as non-zero, it indicates the coding of the macroblock using only a picture reference parameter.

If the field PR_0 does not have a decoded value of zero (codeword '1'), no further information is transmitted for this macroblock. In that case the decoder shall treat the macroblock as an INTER macroblock with the motion vector for the whole block equal to zero, the picture reference parameter equal to PR_0, and with no coefficient data.

If the field PR_0 has a decoded value of zero (codeword '1'), the macroblock is coded. The meaning and usage of the fields MCBPC, CBPB, CBPY, and DQUANT remains unaltered. The field PR is included together with the field MVD for all

INTER macroblocks (and in Improved PB frames mode also for INTRA macroblocks). The use of MODB in Improved PB frames is described in subclause U.3.2.1.4.

U.3.2.1.3 Macroblock Emulation Prevention Bit 0 ($MEPB_0$) (1 bit)

$MEPB_0$ is a single bit fixed length codeword having the value "1" that follows PR_0 if and only if PR_0 is present and has a decoded value of "1" (codeword '000'), and either of the following two conditions are satisfied:

1. the slice structured mode (see Annex K) is in use, or
2. the COD for the current macroblock immediately follows after another macroblock which also has COD = "0" and PR_0 = "1" (codeword '000'), and the PR_0 of the previous macroblock is not followed by an $MEPB_0$ bit.

The purpose of $MEPB_0$ is to prevent start-code emulation and, in the slice structured mode, to aid in determining the number of macroblocks in a slice.

U.3.2.1.4 Macroblock Picture Reference Parameters (PR, PR_{2-4}, and PRB) (variable length)

PR is the primary picture reference parameter. PR is present whenever MVD is present. The three codewords PR_{2-4} are included together with MVD_{2-4} if indicated by PTYPE and if MCBPC specifies an INTER4V or INTER4V+Q macroblock (a macroblock of type 2 or 5 in Tables 8 and 9). PR_{2-4} and MVD_{2-4} are only present when in Advanced Prediction mode (see Annex F) or Deblocking Filter mode (see Annex J). PRB is only present in an Improved PB frame when MODB indicates that MVDB is present. PR, PR_{2-4}, and PRB each specify a picture reference relative index into the multi-picture buffer.

PR is used as the picture reference parameter for motion compensation of the entire macroblock if the macroblock is not an INTER4V or INTER4V+Q macroblock. If the macroblock is an INTER4V or INTER4V+Q macroblock, PR is used for motion compensated prediction of the first of the four 8×8 luminance blocks in the macroblock and for the two chrominance blocks of the macroblock (with the motion compensation process otherwise as specified in subclause 6.1). PR_{2-4} are used for motion compensation of the remaining three 8×8 blocks of luminance data in the macroblock. If MODB indicates that MVDB is present, PRB is the picture reference parameter for forward prediction of the B part of the Improved PB frame.

In Improved PB frames when MODB indicates B_{PB} bidirectional prediction, the values of TR_D and TR_B shall be computed as the temporal reference increments based on the temporal reference data of the current picture and that of the most recent previous reference picture, regardless of whether or not the most recent previous reference picture has been re-mapped to a difference relative index order, marked as "unused", or assigned to a long-term index. The picture used as the forward reference picture for B_{PB} bidirectional prediction in Improved PB frames shall be the picture specified by PR.

U.3.2.1.5 Macroblock Emulation Prevention Bits (MEPB , $MEPB_{2-4}$, and MEPBB) (1 bit each)

MEPB, $MEPB_{2-4}$, and MEPBB are each a single bit having the value "1" if present. Each shall be present if and only if the Unrestricted Motion Vector mode (see Annex D) is not in use and the associated PR, PR_{2-4}, or PRB field is present and has the decoded value "1" (codeword '000'). Their purpose is to prevent start-code emulation.

U.3.2.2 B-Picture and EP-Picture Macroblock Syntax

The macroblock layer syntax for B and EP pictures (see Annex O) is modified in a similar fashion as in P pictures. The COD bit, if equal to "1", indicates a skipped macroblock as defined in Annex O, using a picture reference parameter of zero for the forward (skipped) prediction in an EP picture and for the forward part of direct (skipped) bidirectional prediction in a B picture and using the first backward prediction picture for the backward part of direct (skipped) bidirectional prediction in a B picture (in the case of two-picture backward prediction, as when BSBBW is present and equal to "0"). If COD is "0", a PR_0 parameter is inserted into the syntax and is used in a similar manner as described in U.3.2.1.2. If PR_0 is present and does not have a decoded value of zero (codeword '1'), it indicates that the macroblock is to be predicted with forward INTER prediction using a zero-valued motion vector and a picture reference parameter of PR_0. If PR_0 has a decoded value of zero, MBTYPE follows and specifies the macroblock type. The format of the CBPC, CBPY, and DQUANT fields is unchanged. The MVDFW and MVDBW fields are encoded in the same manner as when the ERPS mode is not in use, but are each used in conjunction with a picture reference, and possibly an emulation prevention bit.

For a B picture, the backward reference pictures in the multi-picture buffer are defined as follows:

- In the case of single-picture backward prediction, there is only one backward reference picture, which is the first picture in (possibly re-mapped) relative index order, and
- In the case of two-picture backward prediction, there are two backward reference pictures, which are the first two pictures in (possibly re-mapped) relative index order.

The forward reference pictures in the multi-picture buffer are defined as the pictures in the multi-picture buffer other than the backward reference pictures. The relative indexing for forward prediction is a relative index into the forward reference picture set, and the relative indexing for backward prediction is a relative index into the backward reference picture set.

For example, if the buffer contains three short-term pictures with short-term picture numbers 300, 302, and 303 (which were transmitted in increasing picture-number order) and two long-term pictures with long-term picture indices 0 and 3, the default index order in the case of two-picture backward prediction is:

- default backward relative index 0 refers to the short-term picture with picture number 303,
- default backward relative index 1 refers to the short-term picture with picture number 302,
- default forward relative index 0 refers to the short-term picture with picture number 300,
- default forward relative index 1 refers to the long-term picture with long-term picture index 0, and
- default forward relative index 2 refers to the long-term picture with long-term picture index 3;

and in the case of single-picture backward prediction:

- the single default backward reference picture is the short-term picture with picture number 303,
- default forward relative index 0 refers to the short-term picture with picture number 302,
- default forward relative index 1 refers to the short-term picture with picture number 300,
- default forward relative index 2 refers to the long-term picture with long-term picture index 0, and
- default forward relative index 3 refers to the long-term picture with long-term picture index 3;

and if these pictures have been re-mapped to a new relative indexing order of short-term picture 302, followed by short-term picture 303, followed by long-term picture 0, followed by short-term picture 300, followed by long-term picture 3, the new relative index order in the case of two-picture backward prediction is:

- re-mapped backward relative index 0 refers to the short-term picture with picture number 302,
- re-mapped backward relative index 1 refers to the short-term picture with picture number 303,
- re-mapped forward relative index 0 refers to the long-term picture with long-term picture index 0,
- re-mapped forward relative index 1 refers to the short-term picture with picture number 300, and
- re-mapped forward relative index 2 refers to the long-term picture with long-term picture index 3;

and in the case of single-picture backward prediction:

- the single re-mapped backward reference picture is the short-term picture with picture number 302,
- re-mapped forward relative index 0 refers to the short-term picture with picture number 303,
- re-mapped forward relative index 1 refers to the long-term picture with long-term picture index 0,
- re-mapped forward relative index 2 refers to the short-term picture with picture number 300, and
- re-mapped forward relative index 3 refers to the long-term picture with long-term picture index 3.

The TR_D used for direct bidirectional prediction in a B picture shall be computed as the temporal reference increment between the first forward reference picture in (possibly re-mapped) relative index order and the first backward reference picture in (possibly re-mapped) relative index order (i.e. if two-picture backward prediction is in use, this would be the picture referenced when BSBBW is "0" as described in sub-clause U.3.2.2.3). The TR_B used for direct bidirectional prediction in a B picture shall be computed as the temporal reference increment between the B picture and the first forward reference picture in (possibly re-mapped) relative index order. The relative index order used in the computation of TR_D and TR_B shall be that specified by the ERPS layer at the picture level of the B picture syntax (i.e. re-mappings at the GOB or slice level shall not affect the values of TR_D and TR_B).

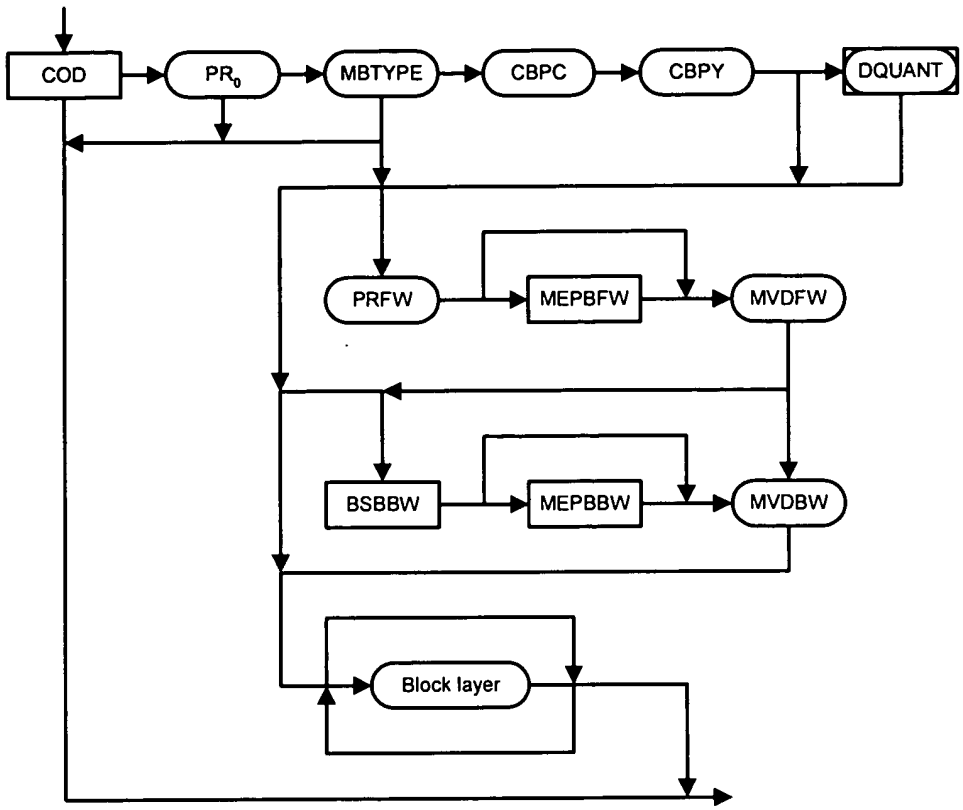


FIGURE U.8/H.263

Structure of EP and B picture Macroblock layer for the ERPS mode

U.3.2.2.1 Picture Reference for Forward Prediction (PRFW) (variable length)
PRFW is a variable length picture reference parameter that is present whenever MVDFW is present, and is encoded using Table U.1. PRFW is a relative index into the set of forward reference pictures.

U.3.2.2.2 Emulation Prevention Bit for Forward Prediction (MEPBFW) (1 bit)
MEPBFW is a single bit fixed length codeword having the value "1" which shall be inserted after PRFW if and only if PRFW is present and has a decoded value of "1" (codeword '000') and the unrestricted motion vector mode (see Annex D) is not in use.

U.3.2.2.3 B-Picture Selection Bit for Backward Prediction (BSBBW) (1 bit)
BSBBW is a single bit fixed length codeword that is present only for B pictures when MVDBW is present and only when two-picture backward prediction is specified for the B picture operation. The meaning of this bit shall be defined as:

"0" : Prediction from the first backward reference picture in relative index order (in default order, this would be the most recent short-term reference picture if that picture has not been assigned a long-term index or marked as "unused")

"1" : Prediction from the second backward reference picture in relative index order (in default order, this would be the second-most recent short-term reference picture if neither of the last two reference pictures has been assigned a long-term index or marked as "unused")

U.3.2.2.4 Emulation Prevention Bit for Backward Prediction (MEPBBW) (1 bit)
MEPBBW is a single bit fixed length codeword having the value "1" that is present only under the following conditions:

- BSBBW is present and equal to "0", and
- The unrestricted motion vector mode (see Annex D) is not in use, and
- BSBBW is preceded by five bits having the value '00000'

U.4 Decoder Process

The decoder for the ERPS mode stores the reference pictures for inter-picture decoding in a multi-picture buffer. The decoder may need additional memory capacity to store the multiple decoded pictures (relative to the memory capacity needed without support of the ERPS mode). The decoder replicates the multi-picture buffer of the encoder according to the reference picture buffering type and any memory management control operations specified in the bitstream. The buffering scheme may also be operated when partially erroneous pictures are decoded.

Each transmitted and stored picture is assigned a Picture Number (PN) which is stored with the picture in the multi-picture buffer. PN represents a sequential picture counting identifier for stored pictures. PN is constrained, using modulo 1024 arithmetic operation. For the first transmitted picture, PN should be "0". For each and every other transmitted and stored picture, PN shall be increased by 1 (within a given scalability layer, if Annex O is in use). If the difference (modulo 1024) of the PNs of two consecutively received and stored pictures is not 1, the decoder should infer a loss of pictures or corruption of data. In such a case, a back-channel message indicating the loss of pictures may be sent to the encoder.

Besides the PN, each picture stored in the multi-picture buffer has an associated index, called the default relative index. When a picture is first added to the multi-picture buffer it is given default relative index 0 – unless it is assigned to a long-term index. The default relative indices of pictures in the multi-picture buffer are modified when pictures are added to or removed from the multi-picture buffer, or when short-term pictures are assigned to long-term indices.

The pictures stored in the multi-picture buffers can also be divided into two categories: long-term pictures and short-term pictures. A long-term picture can stay in the multi-picture buffer for a long time (more than 1023 coded and stored picture intervals). The current picture is initially considered a short-term picture. Any short-term picture can be changed to a long-term picture by assigning it a long-term index according to information in the bitstream. The PN is the unique ID for all short-term pictures in the multi-picture buffer. When a short-term picture is changed to a long-term picture, it is also assigned a long-term picture index (LPIN). A long-term picture index is assigned to a picture by associating its PN to an LPIN. Once a long-term picture index has been assigned to a picture, the only potential subsequent use of the long-term picture's PN within the bitstream shall be in a repetition of the long-term index assignment. The PNs of the long-term pictures are unique within 1024 transmitted and stored pictures. Therefore, the PN of a long-term picture cannot be used for assignment of a long-term index after 1023 transmitted subsequent stored pictures. LPIN becomes the unique ID for the life of a long-term picture.

PN (for a short-term picture) or LPIN (for a long-term picture) can be used to re-map the pictures into re-mapped relative indices for efficient reference picture addressing.

U.4.1 Decoder Process for Short/Long-term Picture Management

The decoder may have both long-term pictures and short-term pictures in its multi-picture buffer. The MLIP1 field is used to indicate the maximum long-term picture index allowed in the buffer. If no prior value of MLIP1 has been sent, no long-term pictures shall be in use, i.e. MLIP1 shall initially have an implied value of "0" upon invocation of the ERPS mode. Upon receiving an MLIP1 parameter, a new MLIP1 shall take effect until another value of MLIP1 is received. Upon receiving a new MLIP1 parameter in the bitstream, all long-term pictures with associated long-term indices greater than or equal to MLIP1 shall be considered marked "unused". The frequency of transmitting MLIP1 is out of the scope of this Recommendation. However, the encoder should send an MLIP1 parameter upon receiving an error message, such as an Intra request message.

A short-term picture can be changed to a long-term picture by using an MMCO command with an associated DPN and LPIN. The short-term picture number is derived from DPN and the long-term picture index is LPIN. Upon receiving such an MMCO command, the decoder shall change the short-term picture with PN indicated by DPN to a long-term picture and shall assign it to the long-term index indicated by LPIN. If a long-term picture with the same long-term index already exists in the buffer, the previously-existing long-term picture shall be marked "unused". An encoder shall not assign a long-term index greater than MLIP1−1 to any picture. If LPIN is greater than MLIP1−1, this condition should be treated by the decoder as an error. For error resilience, the encoder may send the same long-term index assignment operation or MLIP1 specification message repeatedly. If the picture specified in a long-term assignment operation is already associated with the required LPIN, no action shall be taken by the decoder. An encoder shall not assign the same picture to more than one long term index value. If the picture specified in a long-term index assignment operation is already associated with a different long-term index, this condition should be treated as an error. An encoder shall only change a short-term picture to a long-term picture within 1024 transmitted consecutive stored pictures. In other words, a short-term picture shall not stay in the short-term buffer after more than 1023 subsequent stored pictures have been transmitted. An encoder shall not assign a long-term index to a short-term picture that has been marked as "unused" by the decoding process prior to the first such assignment message in the bitstream. An encoder shall not assign a long-term index to a picture number that has not been sent.

U.4.2 Decoder Process for Reference Picture Buffer Mapping

The decoder employs indices when referencing a picture for motion compensation on the macroblock layer using the fields PR_0, PR, PR_2, PR_3, PR_4, PRB, PRFW, and BSBBW. In pictures other than B pictures, these indices are the default relative indices of pictures in the multi-picture buffer when the fields ADPN and LPIR are not present in the current picture, GOB, or slice layer as applicable, and are re-mapped relative indices when these fields are present. In B pictures, the first one or two pictures (depending on BTPSM) in relative index order are used for backward prediction, and the forward picture reference parameters specify a relative index into the remaining pictures for use in forward prediction.

The indices of pictures in the multi-picture buffer can be re-mapped onto newly specified indices by transmitting the RMPNI, ADPN, and LPIR fields. RMPNI indicates whether ADPN or LPIR is present. If ADPN is present, RMPNI specifies the sign of the difference to be added to a picture number prediction value. The ADPN value corresponds to the absolute difference between the PN of the picture to be re-mapped and a prediction of that PN value. The first transmitted ADPN is computed as the absolute difference between the PN of the current picture and the PN of the picture to be re-mapped. The next transmitted ADPN field represents the difference between the PN of the previous picture that was re-mapped using ADPN and that of another picture to be re-mapped. The process continues until all necessary re-mapping is complete. The presence of re-mappings specified using LPIR does not affect the prediction value for subsequent re-mappings using ADPN. If RMPNI indicates the presence of an LPIR field, the re-mapped picture corresponds to a long-term picture with a long-term index of LPIR. If any pictures are not re-mapped to a specific order by RMPNI, these remaining pictures shall follow after any pictures having a re-mapped order in the indexing scheme, following the default order amongst these non-re-mapped pictures.

If the decoder detects a missing picture, it may invoke some concealment process, and may insert an error-concealed picture into the multi-picture buffer. Missing pictures can be identified if one or several picture numbers are missing or if a picture not stored in the multi-picture buffer is indicated in a transmitted ADPN or LPIR. Concealment may be conducted by copying the closest temporally preceding picture that is available in the multi-picture buffer into the position of the missing picture. The temporal order of the short-term pictures in the multi-picture buffer can be inferred from their default relative index order and PN fields. In addition or instead, the decoder may send a forced INTRA update signal to the encoder by external means (for example, Recommendation H.245), or the decoder may use external means or back-channel messages (for example, Recommendation H.245) to indicate the loss of pictures to the encoder. A concealed picture may be inserted into the multi-picture buffer when using the "Sliding Window" buffering type. If a missing picture is detected when decoding a GOB or Slice layer, the concealment may be applied to the picture as if the missing picture had been detected at the picture layer.

U.4.3 Decoder Process for Sub-Picture Removal

Sub-Picture Removal may be used to reduce the amount of memory required to save multiple reference pictures. In sub-picture removal operation, each reference picture is partitioned into smaller equal-sized sub-pictures. The memory reduction is accomplished by marking undesired sub-picture areas as "unused". The strategy used by the encoder to decide which of the sub-pictures to mark as "unused" is outside the scope of this document. The encoder signals to the decoder the size of the sub-pictures and which of the sub-pictures to mark as "unused" using MMCO commands in the enhanced reference picture selection (ERPS) layer. The encoder shall not send information in the bitstream that causes any samples in reference pictures or sub-pictures that it has caused to be marked as "unused" to be indicated for use in the prediction of subsequent pictures.

The sub-picture removal capability is negotiated by external means (for example, Recommendation H.245). In addition, the decoder signals, also by external means, the minimum partition unit (MPU) which is described in terms of a minimum width and height (in units of 16 luminance samples) of a sub-picture and the total amount of memory it has available for its multi-picture buffer. Memory management is facilitated by the partition rules described below.

Each reference picture is partitioned into rectangular sub-pictures of equal size. The encoder specifies the sub-picture size which shall be an integer multiple of the MPU. The width and height of the sub-picture shall be integer multiples of the minimum MPU width and height negotiated externally. The upper-left-hand corner of the first sub-picture is coincident with the upper-left-hand corner of the reference picture. Consequently, the entire partition may be described by specifying the width and height of a sub-picture. If the picture size is not an integer multiple of the sub-picture size, some sub-pictures may extend beyond the right and bottom boundaries of the reference picture if the picture size is not an integer multiple of the sub-picture size. When a sub-picture that extends past the reference picture boundary is saved, a convenient memory management strategy is to set aside enough memory to save the entire sub-picture, rather than just the memory necessary to save the portion of the reference picture that lies within that sub-picture. This is the convention which shall be followed in any calculation of buffer spare capacity for the purpose of determining buffer fullness (e.g. in order to determine whether to automatically mark buffered pictures as "unused" in "sliding window" operation). A decoder designed such that each sub-picture occupies the same amount of memory will prevent the possibility of memory fragmentation.

An example method designed to access referenced picture samples when sub-picture removal is in use is described briefly as follows. One important element in any reference picture access technique is a mechanism to identify where the samples in each sub-picture are stored in memory. If there are R reference pictures and each picture is partitioned into S sub-pictures then there are a total of $K = R \cdot S$ sub-pictures. For example, the sub-picture in the upper-left hand corner of the first reference picture number can be considered sub-picture number 0, and the sub-picture to the right of it can be considered sub-picture number 1 and so on in raster scan order progressing from reference picture 0 to R–1 until all K sub-pictures have a label. The total buffer

capacity is SPTN sub-picture memory buffers, and SPTN is typically less than K. A K-element array can be defined, subPicMem[K], such that t = subPicMem[k] corresponds to the sub-picture memory area that contains the samples in sub-picture k. For example, a case can be considered in which R = 5 reference pictures each have S = 12 sub-pictures. Then the samples for the sub-picture 6 in reference picture 3 would be found in sub-picture memory area t = subPicMem[k] where k = 3·S+6 = 42.

For example, when referencing samples for motion-compensated prediction of one block of luminance or chrominance data when the Advanced Prediction and Reduced Resolution Update optional modes are not in use, it is necessary to acquire n×m samples, where n and m may take values of 8 or 9 to accommodate half-integer motion compensation. Since the samples in one block may lie in up to four different sub-pictures, four separate cases must be considered. In all cases, the first step is to find the location in memory that contains the upper-left hand sample (U) of the block to be referenced. The sub-picture containing U can be identified by dividing the horizontal or vertical location of U by the sub-picture width or height. If U lies in sub-picture k, then that sample will be located in the subPicMem[k] sub-picture memory area. Next, if both the sample m–1 samples to the right of U (i.e. the upper-right-hand corner of the block) and the sample n–1 samples down from U (i.e. the lower-left-hand corner of the block) lie in sub-picture k, this can be considered case number one. If the sample n–1 samples down from U lies within k, but the sample m–1 samples to the right of U does not, this can be considered case two. If the sample m–1 samples to the right of U lies within k, but the sample n–1 down does not, this can be considered case three. Otherwise, when both the sample m–1 samples to the right of U and the one n–1 samples down lie outside of sub-picture k, this can be considered case four.

In case number one, all samples in the reference block are contained within the kth sub-picture. In this case, all relevant n×m samples may be found in sub-picture memory area subPicMem[k]. In case two, the samples that lie in the kth sub-picture can be obtained from sub-picture memory area subPicMem[k] and the remaining samples can be obtained from subPicMem[k_r] where k_r is the sub-picture to the right of k. In case three, the samples that lie in the kth sub-picture can be obtained from memory area subPicMem[k], and the remaining samples can be obtained from subPicMem[k_d] where k_d is the sub-picture below k. In case four, the samples that lie in the kth sub-picture can be obtained from sub-picture memory area subPicMem[k] and the remaining samples can be obtained from memory areas subPicMem[k_r], subPicMem[k_d] and subPicMem[k_{rd}] where k_r and k_d are defined above and k_{rd} is the sub-picture to the right and below k.

U.4.4 Decoder Process for Multi-Picture Motion Compensation

Multi-picture motion compensation is applied if the MRPA field indicates the use of more than one reference picture. For multi-picture motion compensation, the decoder chooses a reference picture as indicated using the fields PR_0, PR, PR_2, PR_3, PR_4, PRB, PRFW, PRBW, and BSBBW on the macroblock layer. Once, the reference picture is specified, the decoding process for motion compensation proceeds as described in subclause 6.1.

In case four motion vectors per macroblock are used and the MRPA field indicates the use of more than one reference picture, the picture reference index for both chrominance blocks is that associated with the first of the four motion vectors (with the motion compensation process otherwise as specified by subclause 6.1).

U.4.5 Decoder Process for Reference Picture Buffering

The buffering of the currently decoded picture can be specified using the reference picture buffering type (RPBT) for non-B pictures. The buffering may follow a first-in, first-out ("Sliding Window") mode. Alternatively, the buffering may follow a customized adaptive buffering ("Adaptive Memory Control") operation that is specified by the encoder in the forward channel. B pictures do not affect buffer contents.

The "Sliding Window" buffering type operates as follows. First, the decoder determine whether the picture can be stored into "unused" buffer capacity. If there is insufficient "unused" buffer capacity, the short term picture with the largest default relative index (i.e. the oldest short-term picture in the buffer) shall be marked as "unused". This process is repeated if necessary (in the case of sub-picture removal) until sufficient memory capacity is freed to hold the current decoded picture. The current picture is stored in the buffer and assigned a default relative index of zero. The default relative index of all other short-term pictures is incremented by one. The default relative indices of all long-term picture are incremented by one minus the number of short-term pictures removed.

In the "Adaptive Memory Control" buffering type, specified pictures or sub-picture areas may be removed from the multi-picture buffer explicitly. The currently decoded picture, which is initially considered a short-term picture, may be inserted into the buffer with default relative index 0, may be assigned to a long-term index, or may be marked as "unused" by the encoder. Other short-term pictures may also be assigned to long-term indices. The buffering process shall operate in a manner functionally equivalent to the following: First, the current picture is added to the multi-picture buffer with default relative index 0, and the default relative indices of all other pictures are incremented by one. Then, the MMCO commands are processed:

- If MMCO indicates a reset of the buffer contents by using RESET equal to "1", all pictures in the buffer are marked as "unused" except the current picture (which will be the picture with default relative index 0 since a buffer reset must be the first MMCO command as required by subclause U.3.1.5.7).

- If MMCO indicates a maximum long-term index using MLIP1, all long-term pictures having long-term indices greater than or equal to MLIP1 are marked as "unused" and the default relative index order of the remaining pictures are not affected.
- If MMCO indicates that a picture is to be marked as "unused" in the multi-picture buffer and if that picture has not already been marked as "unused", the specified picture is marked as "unused" in the multi-picture buffer and the default relative indices of all subsequent pictures in default order are decremented by one.
- If MMCO indicates that sub-picture areas of some picture are to be marked as "unused" in the multi-picture buffer, the specified sub-picture areas are marked as "unused" and the default relative index order of the pictures is not affected. As required by subclause, U.3.1.5.10, not all sub-picture areas of any given picture will be marked "unused" by a sub-picture removal MMCO command (instead, the encoder should send an MMCO command marking the picture as a whole as "unused").
- If MMCO indicates the assignment of a long-term index to a specified short-term picture and if the specified long-term index has not already been assigned to the specified short-term picture, the specified short-term picture is marked in the buffer as a long-term picture with the specified long-term index. If another picture is already present in the buffer with the same long-term index as the specified long-term index, the other picture is marked as "unused". All short-term pictures that were subsequent to the specified short-term picture in default relative index order and all long-term pictures having a long-term index less than the specified long-term index have their associated default relative indices decremented by one. The specified picture is assigned to a default relative index of one plus the highest of the incremented default relative indices, or zero if there are no such incremented indices.

The resulting buffered quantity of pictures or sub-picture regions not marked as "unused" shall not exceed the buffer capacity indicated by the most recent value of SPTN. If the decoder detects this condition, it should be treated as an error.

U.5 Back-Channel Messages

An out-of-band channel, which need not necessarily be reliable, can be used to convey back-channel messages. The syntax of this out-of-band channel (which could be a separate logical channel, for example using Recommendation H.223 or Recommendation H.225.0) should be the one defined herein. The "videomux" operation of back-channel messages as defined in Annex N is not supported in the ERPS mode.

U.5.1 BCM Separate Logical Channel Layer

The BCM layer as specified in subclause U.5.2 should be carried by a BCM Separate Logical Channel layer as shown in Figure U.9.

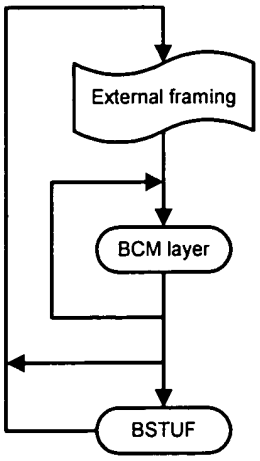


FIGURE U.9/H.263

Structure of BCM Separate Logical Channel layer for ERPS mode

U.5.1.1 External Framing

External framing of back-channel messages should be provided as shown in Figure U.9. The external framing is used to determine the starting point for the back-channel messages and the amount of back-channel message data to follow.

U.5.1.2 Back-Channel Stuffing (BSTUF) (variable length)

BSTUF is a variable length codeword that may be present only after the last back-channel message in an external frame. BSTUF consists of a codeword of variable length consisting of one or more bits of value "0".

U.5.2 Back-Channel Message Layer Syntax

The syntax for the back-channel message (BCM) layer defined herein shall be as shown in Figure U.10.

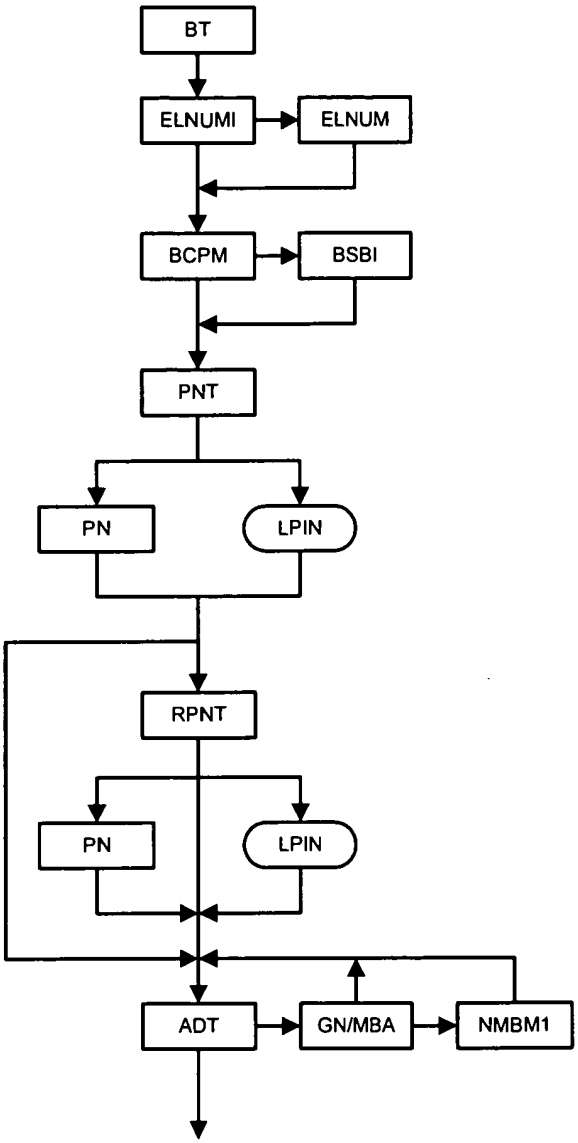


FIGURE U.10/H.263

Structure of Back-Channel Message (BCM) layer for ERPS mode

U.5.2.1 Back-Channel Message Type (BT) (2 bits)

BT is a two bit fixed length codeword which indicates the type of back-channel message. BT is the first codeword present in each back-channel message. Which type or types of message are requested by the encoder is indicated in the RPSMF field of the forward-channel syntax. The values of BT shall be defines as:

"00": Reserved for future use,

"01": Reserved for future use,

"10": NACK: This indicates the loss or erroneous decoding of the corresponding part of the forward channel data,
"11": ACK: This indicates the correct decoding of the corresponding part of the forward channel data.

U.5.2.2 Enhancement Layer Number Indication (ELNUMI) (1 bit)
ELNUMI is a single bit fixed length codeword that follows BT in the back-channel message. ELNUMI shall be "0" unless the optional Temporal, SNR, and Spatial Scalability mode (see Annex O) is used in the forward channel and some enhancement layers of the forward channel are combined in one logical channel and the back-channel message refers to an enhancement layer (rather than the base layer), in which case ELNUMI shall be "1".

U.5.2.3 Enhancement Layer Number (ELNUM) (4 bits)
ELNUM is a 4 bit fixed length codeword that is present only if ELNUMI is "1". It follows ELNUMI if present. When present, ELNUM contains the layer number of the enhancement layer referred to in the back-channel message.

U.5.2.4 Back-Channel CPM Indicator (BCPM) (1 bit)
BCPM is a single bit fixed length codeword that follows ELNUMI or ELNUM in the back-channel message. BCPM shall be "0" unless the CPM mode (see subclause 5.2.4 and Annex C) is used in the forward channel data, in which case BCPM shall be "1". If BCPM is "1", this indicates that BSBI is present.

U.5.2.5 Back-Channel Sub-Bitstream Indicator (BSBI) (2 bits)
BSBI is a 2 bit fixed length codeword that follows BCPM when present. BCPM is present only if BCPM is "1". BSBI is the natural binary representation of the Sub-Bitstream number in the forward channel data to which the back-channel message refers (see subclause 5.2.4 and Annex C).

U.5.2.6 Picture Number Type (PNT) (1 bit)
PNT is a single bit fixed length codeword that is always present and follows BCPM or BSBI in the back-channel message. The values of PNT shall be defined as:
"0": The message concerns a picture specified by a short-term picture number (PN),
"1": The message concerns a picture specified by a long-term picture index (LPIN).
PNT is followed by PN or LPIN, depending on the value of PNT. PN and LPIN shall be represented as specified for use in forward channel data in subclauses U.4.1.3 and U.4.1.5.9, respectively.

U.5.2.7 Requested Picture Number Type (RPNT) (2 bits)
RPNT is a 2 bit fixed length codeword that is present only if BT indicates a NACK message. It follows PN or LPIN when present. It determines how to identify a picture in the multi-picture buffer which may be used as a reference for the coding of subsequent pictures. The values of RPNT shall be defined as:
"00": No valid pictures in buffer – buffer should be reset by an I or EI picture with RESET equal to "1",
"01": No particular picture is identified to be used as a reference,
"10": A picture which may be used as a reference is identified by a short-term picture number (PN),
"11": A picture which may be used as a reference is identified by a long-term picture index (LPIN).

If RPNT is "10" or "11", RPNT is followed by PN or LPIN, depending on the value of RPNT. PN and LPIN shall be represented as specified for use in forward channel data in subclauses U.4.1.3 and U.4.1.5.9, respectively. Typically the PN or LPIN specified using RPNT identifies the last correctly decoded spatially-corresponding picture area for the picture or region identified in the back-channel message.

U.5.2.8 Additional Data Type (ADT) (2 bits)
ADT is a 2 bit fixed length codeword that is present after PN, LPIN, or RPNT, as determined by PNT (in an ACK message) or RPNT (in a NACK message). It may occur multiple times if present. It specifies the type of additional data used to identify a region of the picture of concern to which the back-channel message applies. The values of ADT shall be defined as:
"00": End of additional data,
"01": A region is identified by only a GN/MBA field,
"10": A region is identified as a raster-scan area within a picture by GN/MBA and NMBM1,
"11": A region is identified as a raster-scan area within a rectangular slice by GN/MBA and NMBM1.

If ADT is "00", no more data follows in the back-channel message. If ADT is "01", ADT is followed by GN/MBA and then by another ADT. If ADT is "10" or "11", ADT is followed by GN/MBA and NMBM1 and then by another ADT.

If ADT is "10", the region is identified as a region starting at a particular spatial location specified by GN/MBA and containing a specified number of macroblocks in raster-scan order within the picture. If ADT is "11", the region is identified as a region starting at a particular spatial location specified by GN/MBA and containing a specified number of macroblocks in raster-scan order within a rectangular slice. If ADT is present only once and is "00", the region identified is the picture as a whole. If ADT is present more than once, the value "00" is used only to end the loop rather than to identify a region.

U.5.2.9 GOB Number/Macroblock Address (GN/MBA) (5/6/7/9/11/12/13/14 bits)

GN/MBA is a fixed length codeword which specifies a GOB number or macroblock address. GN/MBA follows ADT when present. GN/MBA is present when indicated by ADT. If the optional Slice Structured mode (see Annex K) is not in use, GN/MBA contains the GOB number of the beginning of an area to which the back-channel message refers. If the optional Slice Structured mode is in use, GN/MBA contains the macroblock address of the beginning of the area to which the back-channel message refers. The length of this field shall be as specified elsewhere in this Recommendation for GN or MBA.

U.4.5.2.10 Number of Macroblocks Minus 1 (NMBM1) (5/6/7/9/11/12/13/14 bits)

NMBM1 is a fixed length codeword which specifies a number of macroblocks. NMBM1 is present when indicated by ADT. It follows GN/MBA when present. It contains the natural representation of the number of specified macroblocks minus 1. The length of this field shall be the length defined for a macroblock address in subclause K.2.5 and Table K.2.

Annex V
Data Partitioned Slice Mode

(This annex forms an integral part of this Recommendation.)

V.1 Introduction

This annex describes the optional data-partitioned slice (DPS) mode of H.263. The capability of this mode is signaled by external means (for example Recommendation H.245). The use of this mode shall be indicated by setting the formerly-reserved bit 17 of the optional part of the PLUSPTYPE (OPPTYPE) to '1'. This mode uses the header structure defined in Annex K.

Data partitioning provides robustness in error prone environments. This is accomplished using a rearrangement of the H.263 syntax to enable early detection of and recovery from errors that have been introduced during transmission.

V.2 Structure of data partitioning

When data partitioning is used, the data is arranged as a video picture segment, as defined in Section R.2. The MB's in the segment are rearranged so that the header information for all the MB's in the segment are transmitted together, followed by the MV's for all the MB's in the segment, and then by the DCT coefficients for all the MB's in the segment. The segment header uses the same syntax as described in Section K.2. The header, MV, and DCT partitions are separated by markers, allowing for resynchronization at the end of the partition in which an error occurred. Each segment shall contain the data for an integer number of MB's. When this mode is in use the syntax shown in Figure V.1 shall be used.

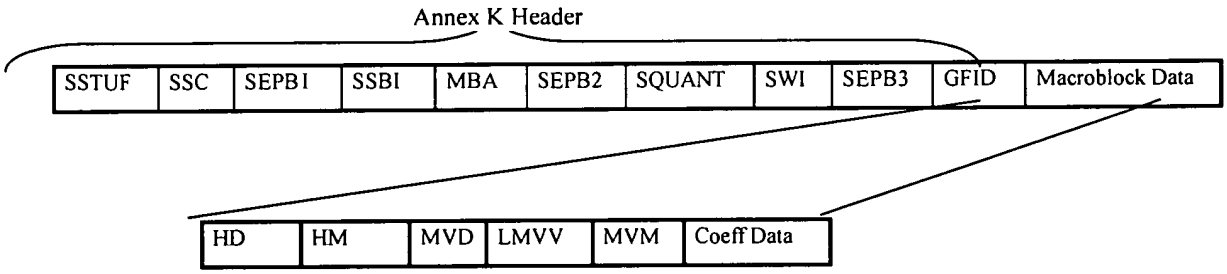


FIGURE V.1/H.263

Data Partitioning Syntax

Note that when this annex is not active, the MV and DCT data are transmitted in an interleaved fashion for all the MB's in a video picture segment, in which case an error normally results in the loss of all information for the remaining MB's in the packet.

V.2.1 Header Data (HD) (Variable length)

The Header Data field contains the COD and MCBPC information for all the MB's in the packets, plus the MODB data in case of PB-frames or Improved PB-frames. A reversible variable length code (RVLC) is used to combine the COD and the MCBPC for all the MB's in the packet. This code is shown in Tables V.1 through V.5/H.263. If Annex O is in use, the COD is only combined with the MB TYPE to form the RVLC for B and EP pictures using tables V.3 and V.4, and the CBPC is coded with codewords in Table O.4. If COD=0 and Annex G or Annex M is in use, the codeword for the COD+MCBPC shall be immediately followed by the reversible variable-length encoded data corresponding to the MODB field of the macroblock. Table V.6 shall be used for PB-frames, Table V.7 shall be used for Improved PB-frames.

V.2.2 Header Marker (HM) (9 bits)

A codeword of 9 bits. Its value is 1010 0010 1. The HM terminates the header partition. When reversed decoding is used by a decoder, the decoder searches for this marker. This value cannot occur naturally in the HD field.

V.2.3 Motion Vector Data Layer(Variable length)

V.2.3.1 Motion Vector Difference Coding

For the motion vectors, the RVLC codewords shown in Table D.3/H.263 are used to encode the difference between the motion vector and the motion vector prediction. Note that this annex only uses the entropy coding from Annex D, but not the other aspects of it unless Annex D is also in use.

V.2.3.2 Prediction of Motion Vector Values

The first motion vector in the packet is coded using a predictor value of 0 for both horizontal and vertical components, and the MV's for the subsequent coded MB's are coded predictively using the MV difference (MVD). This differs from the method otherwise used for coding the MV's in which the MV's following a skipped or INTRA MB are coded using a predictor value of 0 for both horizontal and vertical components.

Forward Direction: $MV_i = MV_{i-1} + MVD_i = MV_{i-1} + (MV_i - MV_{i-1})$

Backward Direction: $MV_{i-1} = MV_i - MVD_i = MV_i - (MV_i - MV_{i-1})$.

(MV_i and MVD_i are the *i*th MV and MV Difference in the packet respectively)

The motion vector information for the last motion vector in the packet is coded in this manner and is also coded again in the LMVV field as described below in V.2.4. This allows the decoder to independently decode the sequence of MV's using two different prediction paths: 1) in the forward direction, starting from the beginning of the motion data of the packet, and 2) in the backward direction, from the end of the motion data in a packet. This provides robustness for better error detection and concealment.

NOTE: When the DPS mode is not in use, motion vectors are predictively coded, with the prediction of the current motion vector being the median value of 3 motion vectors of neighboring locations as described in Section 6.1.1. Because packets in this annex are formed in a way such that the number of MB's coded in each packet is variable, using the median predictive coding method (which involves motion vectors on different rows of the frame) would prevent reversible decoding of the motion vectors in a slice. When the DPS mode is in use, a single prediction thread is formed for the MV's in the whole packet. This is shown in Figure V.2.

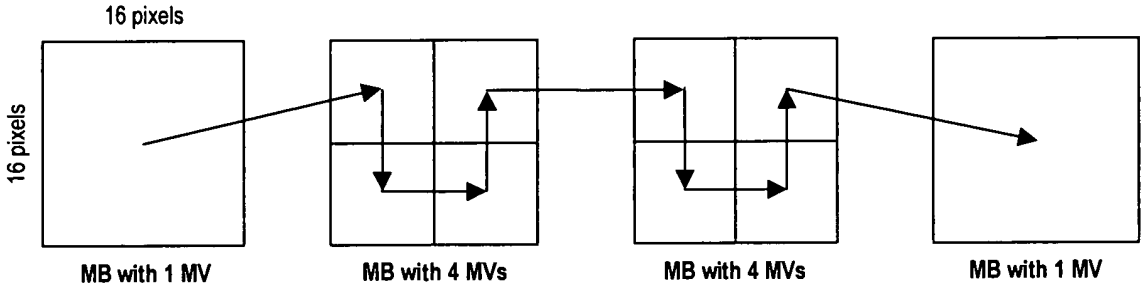


FIGURE V.2/H.263

Single Thread Motion Vector Prediction

In case of B pictures or EP pictures (Annex O), MVDFW and MVDBW may be present as indicated by the MBTYPE codeword in tables V.3 and V.4. MVDFW is predictively encoded using the same single prediction thread as described above and MVDBW (when present in B pictures) shall be encoded as specified in O.4.6. MVDFW and MVDBW shall be coded with the codewords from Table D.3/H.263.

In case of PB-frames (Annex G) and Improved PB-frames (Annex M), the MVDB data shall be encoded as specified in corresponding annexes and shall be coded using the codewords from Table D.3/H.263.

NOTE – If the backward decoding mode is engaged in a B frame (Annex O) or in Improved PB-frames (Annex M), MVDB and MVDBW should be discarded by the decoder as the Motion Vector data for the backward prediction may not be recovered properly across the packet boundaries.

V.2.3.3 Start-Code Emulation Prevention in Motion Vector Difference Coding

The MVD start-code-emulation avoidance method is changed from the method described in Section D.2 of Annex D, in order to facilitate independent parsing in the backward direction. A MVD=0 (codeword "1") shall be inserted between any two consecutive MVD's that are both equal to 1 (codeword "000"). This differs from Annex D, in which the bit is only inserted when two consecutive MVD=1 form a pair (i.e. when the first MVD is the horizontal component, and the second is the vertical component). If Annex D and Annex V are both in use, this Annex V method of start-code-emulation avoidance method shall be used instead of the method described in Section D.2.

V.2.4 Last Motion Vector Value (LMVV) (Variable length)

The LMVV field contains the last MV in the packet. It is coded using a predictor value of 0 for both the horizontal and vertical components. If there are no motion vectors or only one motion vector in the packet, LMVV shall not be present. (This use of a fixed zero-valued predictor enables the use of reversible decoding.)

V.2.5 Motion Vector Marker (MVM) (10 bits)

A codeword of 10 bits having the value '0000 0000 01'. The MVM terminates the motion vector partition. When reverse decoding is used in a decoder, the decoder searches for this marker. The Motion Vector Marker (MVM) shall not be included in the packet if the packet does not contain Motion Vector Data (if all the macroblocks in the packet are intra-coded or with COD's equal to 1).

V.2.6 Coefficient Data Layer (Variable length)

The DCT data layer contains INTRA_MODE (if present), CBPB (if present), CBPC (if present), CBPY, DQUANT (if present), and DCT coefficients coded as specified in Sections I.2, 5.3.4, O.4.3, 5.3.5, 5.3.6, and 5.4.2, respectively. The syntax diagram of DCT Data is illustrated in Figure V.3. The presence of CBPC is indicated in tables V.3 and V.4.

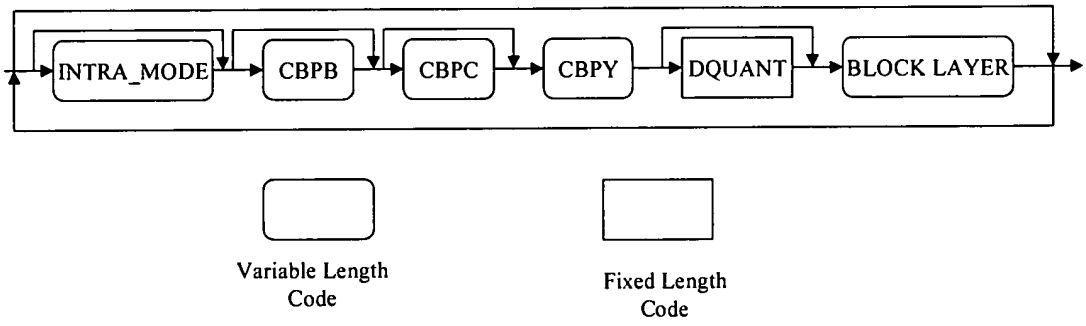


FIGURE V.3/H.263

Coefficient Data syntax

V.3 Interaction with Other Optional Modes

The DPS mode acts effectively as a sub-mode of the Slice Structured mode of Annex K, and uses its outer picture and slice header structures. The SS mode shall therefore be indicated as being in use whenever the DPS mode is in use. Both of the other sub-modes of the Slice Structured mode (the Arbitrary Slice Ordering and Rectangular Slice sub-modes) may be used in conjunction with the DPS mode.

The Syntax-Based Arithmetic Coding mode of Annex E shall not be used with this annex, as it does not allow for reversible decoding.

Annex H Forward Error Correction should not be used with this annex, as it can result in the bitstream being disrupted in undesirable places. However, the use of Annex H with the DPS mode is not forbidden, as the FEC defined in Annex H is required in some existing standard system designs.

The Temporal, SNR, and Spatial Scalability (TSSS) mode of Annex O may be used in conjunction with the DPS mode. When the TSSS and DPS modes are used together, the codewords provided in Tables V.3, V.4, and V.5 shall be used instead of those defined in Annex O.

Annex U shall not be used with this Annex.

TABLE V.1/H.263

COD + MCBPC RVLC table for INTRA MB's

MB type	CBPC (56)	Codeword (for combined COD+MCBPC)	Number of Bits
3 (INTRA)	00	1	1
3	01	010	3
3	10	0110	4
3	11	01110	5
4 (INTRA+Q)	00	00100	5
4	01	011110	6
4	10	001100	6
4	11	0111110	7
stuffing		0011100	7

TABLE V.2/H.263

COD + MCBPC RVLC Table for INTER MB's

MB type	CBPC (56)	Codeword (for combined COD+MCBPC)	Number of Bits
skipped		1	1
0 (INTER)	00	010	3
0	10	00100	5
0	01	011110	6
0	11	0011100	7
1 (INTER + Q)	00	01110	5
1	10	00011000	8
1	01	011111110	9
1	11	01111111110	11
2 (INTER4V)	00	0110	4
2	10	01111110	8
2	01	00111100	8
2	11	000010000	9
3 (INTRA)	00	001100	6
3	11	0001000	7
3	10	001111100	9
3	01	000111000	9
4 (INTRA + Q)	00	0111110	7
4	11	0011111100	10
4	10	0001111000	10
4	01	0000110000	10
5 (INTER4V + Q)	00	00111111100	11
5	01	00011111000	11
5	10	00001110000	11
5	11	00000100000	11
stuffing		0111111110	10

TABLE V.3/H.263

MBTYPE RVLC codes for B MB's

Index	Prediction Type	MVDFW	MVDBW	CBPC + CBPY	DQUANT	MBTYPE	Bits
—	Direct (skipped)					1 (COD=1)	1
0	Direct			X		010	3
1	Direct + Q			X	X	001100	6
2	Forward (no texture)	X				00100	5
3	Forward	X		X		011110	6
4	Forward + Q	X		X	X	01111110	8
5	Backward (no texture)		X			0110	4
6	Backward		X	X		01110	5
7	Backward + Q		X	X	X	00111100	8
8	Bi-Dir (no texture)	X	X			0011100	7
9	Bi-Dir	X	X	X		0001000	7
10	Bi-Dir + Q	X	X	X	X	0111110	7
11	INTRA			X		00011000	8
12	INTRA + Q			X	X	011111110	9
13	Stuffing					001111100	9

TABLE V.4/H.263

MBTYPE RVLC Table for EP MB's

Index	Prediction Type	MVDFW	MVDBW	CBPC + CBPY	DQUANT	MBTYPE	Bits
—	Forward (skipped)					1 (COD=1)	1
0	Forward	X		X		010	3
1	Forward + Q	X		X	X	0110	4
2	Upward (no texture)					01110	5
3	Upward			X		00100	5
4	Upward + Q			X	X	011110	6
5	Bi-Dir (no texture)					001100	6
6	Bi-Dir	X		X		0111110	7
7	Bi-Dir + Q	X		X	X	0011100	7
8	INTRA			X		0001000	7
9	INTRA + Q			X	X	01111110	8
10	Stuffing					00111100	8

TABLE V.5/H.263

COD + MCBPC RVLC Table for EI MB's

Prediction type	QCBP (56)	Codeword (for combined COD+MCBPC)	Number of Bits
Upward (skipped)		1	1
0 (Upward)	00	010	3
0	01	0110	4
0	10	01110	5
0	11	00100	5
1 (Upward + Q)	00	011110	6
1	01	001100	6
1	10	0111110	7
1	11	0011100	7
2 (INTRA)	00	0001000	7
2	01	01111110	8
2	10	00111100	8
2	11	00011000	8
3 (INTRA + Q)	00	011111110	9
3	01	001111100	9
3	10	000111000	9
3	11	000010000	9
Stuffing		0111111110	10

TABLE V.6/H.263

RVLC Table for MODB

Index	CBPB	MVDB	Number of bits	Code
0			3	010
1		x	4	0110
2	x	x	5	01110

Note: "x" means that the item is present in the macroblock

TABLE V.7/H.263

RVLC Table for MODB for Improved PB-frames mode

Index	CBPB	MVDB	Number of bits	Code	Coding Mode
0			3	010	Bi-directional prediction
1	x		4	0110	Bi-directional prediction
2		x	5	01110	Forward prediction
3	x	x	5	00100	Forward prediction
4			6	011110	Backward prediction
5	x		6	001100	Backward prediction

Note — The symbol "x" in the table above indicates that the associated syntax element is present.

Annex W
Additional Supplemental Enhancement Information Specification

(This annex forms an integral part of this Recommendation.)

W.1 Introduction

This annex describes the format of the additional supplemental enhancement information sent in the PSUPP field of the picture layer of H.263, which adds to the functionality defined in Annex L. The capability of a decoder to provide any or all of the capabilities described in this annex may be signaled by external means (for example, Recommendation H.245). Decoders which do not provide the additional capabilities may simply discard any of the newly defined PSUPP information bits that appear in the bitstream. The presence of this supplemental enhancement information is indicated by the presence of both the PEI bit, and by the following PSUPP octet whose FTYPE field has one of the two newly defined values. The basic interpretation of PEI, PSUPP, FTYPE, and DSIZE is identical to Annex L and to sections 5.1.24 and 5.1.25.

W.2 References

The following Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below.

[8] ISO/IEC 10646-1 (1993): Universal Multiple Octet Coded Character Set

[9] IETF RFC 2396 (1998): Uniform Resource Identifiers (URI): Generic Syntax

W.3 Additional FTYPE Values

Two values that were reserved in Annex L, Table L.1 are defined as follows.

TABLE W.1/H.263

FTYPE Function Type Values

13	Fixed-Point IDCT
14	Picture Message

W.4 Recommended Maximum Number of PSUPP Octets

When using any of the aforementioned FTYPE functions defined in this annex, the total number of PSUPP octets per picture should, in relation to the coded picture size, be kept reasonably small, and should not exceed 256 octets regardless of the coded picture size.

NOTE: Some data transmission protocols used for conveyance of the video bitstream may provide for external repetition of picture header contents for error resilience purposes, and may place limits on the amount of such data that can be repeated from a picture header (e.g., 504 bits in the IETF RFC 2429 packetization format). The inclusion of a large number of PSUPP octets may result in the lack of such an external protocol to provide for full repetition of the picture header contents.

W.5 Fixed-Point IDCT

The fixed-point IDCT function indicates that a particular IDCT approximation is used in construction of the bitstream. DSIZE shall be equal to 1 for the fixed-point IDCT function. The octet of PSUPP data that follows specifies the particular IDCT implementation. A value of 0 indicates the reference IDCT 0 as described in W.5.3; values of 1 through 255 are reserved.

W.5.1 Decoder Operation

The capability of a decoder to perform a particular fixed-point IDCT may be signaled to the encoder by external means (for example, Recommendation H.245). When receiving an encoded bitstream with the fixed-point IDCT indication, a decoder shall use the particular fixed-point IDCT if it is capable of doing so.

W.5.2 Removal of Forced Updating

Annex A specifies the accuracy requirements for the inverse discrete cosine transform (IDCT), allowing numerous compliant implementations. To control accumulation of errors due to mismatched IDCTs at the encoder and decoder, Section 4.4 Forced Updating requires that macroblocks be coded in INTRA mode at least once every 132 times when coefficients are transmitted.

If the fixed-point IDCT function type is indicated in the bitstream, then the forced updating requirement is removed, and the frequency of INTRA coding is unregulated. An encoder should continue to use forced updating, however, unless it has ascertained through external means that the decoder is capable of the particular fixed-point IDCT specified herein; otherwise there may be mismatch.

W.5.3 Reference IDCT 0

The reference IDCT 0 is any implementation that, for every input block, produces identical output values as the C source program listed below.

NOTE: This fixed-point IDCT is compliant with Annex A of ITU-T Recommendation H.263, but is not compliant with the extended range of values requirement in Annex A of ITU-T Recommendation H.262 | ISO/IEC 13818-2.

```
/*****************************************************************************
 *
 *                            FIXED-POINT IDCT
 *
 * Fixed-point fast, separable idct
 * Storage precision: 16 bits signed
 * Internal calculation precision: 32 bits signed
 * Input range: 12 bits signed, stored in 16 bits
 * Output range: [-256, +255]
 * All operations are signed
 *
 *****************************************************************************/

/*
 * Includes
 */

#include <stdlib.h>
#include <stdio.h>

/*
 * Typedefs
 */

typedef short int REGISTER;  /* 16 bits signed */
typedef long int LONG; /* 32 bits signed */

/*
 * Global constants
 */

const REGISTER cpo8   = 0x539f;     /* 32768*cos(pi/8)*1/sqrt(2)    */
const REGISTER spo8   = 0x4546;     /* 32768*sin(pi/8)*sqrt(2)    */
const REGISTER cpo16  = 0x7d8a;     /* 32768*cos(pi/16)   */
const REGISTER spo16  = 0x18f9;     /* 32768*sin(pi/16)   */
const REGISTER c3po16 = 0x6a6e;     /* 32768*cos(3*pi/16) */
const REGISTER s3po16 = 0x471d;     /* 32768*sin(3*pi/16) */
const REGISTER OoR2   = 0x5a82;     /* 32768*1/sqrt(2)    */

/*
 * Function declarations
 */

void Transpose(REGISTER block[64]);
void HalfSwap(REGISTER block[64]);
void Swap(REGISTER block[64]);
void Scale(REGISTER block[64], signed char sh);
void Round(REGISTER block[64], signed char sh,
          const REGISTER min, const REGISTER max);
REGISTER Multiply(const REGISTER a, REGISTER x, signed char sh);
void Rotate(REGISTER *x, REGISTER *y,
            signed char sha, signed char shb,
            const REGISTER a, const REGISTER b,
            int inv);
void Butterfly(REGISTER column[8], char pass);
void IDCT(REGISTER block[64]);

/*
 * Transpose():
 *      Transpose a block
```

```
 * Input:
 *      REGISTER block[64]
 * Output:
 *      block
 * Return value:
 *      none
 */
void Transpose(REGISTER block[64])
{
  int i, j;
  REGISTER temp;

  for (i=0; i<8; i++) {
    for (j=0; j<i; j++) {
      temp = block[8*i+j];
      block[8*i+j] = block[8*j+i];
      block[8*j+i] = temp;
    }
  }
  return;
}

/*
 * HalfSwap():
 *      One-dimensional swap
 * Input:
 *      REGISTER block[64]
 * Output:
 *      block
 * Return value:
 *      none
 */
void HalfSwap(REGISTER block[64])
{
  int i;
  REGISTER temp;

  for (i=0; i<8; i++) {
    temp = block[8+i];
    block[8+i] = block[32+i];
    block[32+i] = temp;
    temp = block[24+i];
    block[24+i] = block[48+i];
    block[48+i] = temp;
    temp = block[40+i];
    block[40+i] = block[56+i];
    block[56+i] = temp;
  }
  return;
}

/*
 * Swap():
 *      Swap and transpose a block
 * Input:
 *      REGISTER block[64]
 * Output:
 *      block
 * Return value:
 *      none
 */
void Swap(REGISTER block[64])
{
  HalfSwap(block);
  Transpose(block);
  HalfSwap(block);
}

/*
 * Scale():
 *      Scale a block
 * Input:
 *      REGISTER block[64]
 *      signed char sh
 * Output:
 *      block
 * Return value:
```

```
 *      none
 */
void Scale(REGISTER block[64], signed char sh)
{
  int i;

  if (sh>0) {
    for (i=0; i<64; i++)
      block[i] >>= sh;
  }
  else {
    for (i=0; i<64; i++)
      block[i] <<= -sh;
  }
}

/*
 * Round():
 *      Performs the final rounding of an 8x8 block
 * Input:
 *      REGISTER block[64]
 *      signed char sh
 *      const REGISTER min
 *      const REGISTER max
 * Output:
 *      block
 * Return value:
 *      none
 */
void Round(REGISTER block[64], signed char sh,
           const REGISTER min, const REGISTER max)
{
  int i;

  for (i=0; i<64; i++) {
    if (block[i] < 0x00007FFF - (1<<(sh-1)))
      block[i] += (1<<(sh-1));
    else
      block[i] = 0x00007FFF;
    block[i] >>= sh;
    block[i] = (block[i]<min) ? min : ((block[i]>max) ? max : block[i]);
  }
  return;
}

/*
 * Multiply():
 *      Multiply by a constant with shift
 * Input:
 *      const REGISTER a
 *      REGISTER x
 *      signed char sh
 * Output:
 *      none
 * Return value:
 *      REGISTER, the result of the multiply
 */
REGISTER Multiply(const REGISTER a, REGISTER x, signed char sh)
{
  LONG tmp;
  REGISTER reg_out;

  /* multiply */
  tmp = (LONG)a * (LONG)x;

  /* shift */
  if (sh > 0)
    tmp >>= sh;
  else
    tmp <<= -sh;

  /* rounding and saturating */
  if (tmp < 0x7FFFFFFF - 0x00007FFF)
    tmp = tmp + 0x00007FFF;
  else
    tmp = 0x7FFFFFFF;
```

```
  reg_out = (REGISTER)(tmp >>16);

  return(reg_out);
}

/*
 * Rotate():
 *      Perform rotate operation on two registers
 * Input:
 *      REGISTER *x         pointer to the 1st register
 *      REGISTER *y         pointer to the 2nd register
 *      signed char sha     shift associated with factor a
 *      signed char shb     shift associated with factor b
 *      const REGISTER a    factor a
 *      const REGISTER b    factor b
 *      int inv             1 for inverse dct, 0 for forward dct
 * Output:
 *      *x, *y
 * Return value:
 *      none
 */
void Rotate(REGISTER *x, REGISTER *y,
            signed char sha, signed char shb,
            const REGISTER a, const REGISTER b,
            int inv)
{
  LONG tmplxa, tmplya, tmplxb, tmplyb;
  LONG tmpl1, tmpl2;

  /*
   * intermediate calculation
   */

  tmplxa = (LONG)(*x) * (LONG)a;
  if (sha > 0)
    tmplxa >>= sha;
  else
    tmplxa <<= -sha;

  tmplya = (LONG)(*y) * (LONG)a;
  if (sha > 0)
    tmplya >>= sha;
  else
    tmplya <<= -sha;

  tmplxb = (LONG)(*x) * (LONG)b;
  if (shb > 0)
    tmplxb >>= shb;
  else
    tmplxb <<= -shb;

  tmplyb = (LONG)(*y) * (LONG)b;
  if (shb > 0)
    tmplyb >>= shb;
  else
    tmplyb <<= -shb;

  /*
   * rounding and rotation
   */

  if (inv) {
    tmplxa += 0x00007FFF;
    tmplxb += 0x00007FFF;

    tmpl1 = tmplxb - tmplya;
    tmpl2 = tmplxa + tmplyb;
  }
  else {
    tmplya += 0x00007FFF;
    tmplyb += 0x00007FFF;

    tmpl1 =  tmplxb + tmplya;
    tmpl2 = -tmplxa + tmplyb;
  }
```

```
  /*
   * final rounding
   */

  *x = (REGISTER) (tmp11 >>16);
  *y = (REGISTER) (tmp12 >>16);

  return;
}

/*
 * Butterfly():
 *      Perform 1D IDCT on a column
 * Input:
 *      REGISTER column[8]
 *      char pass
 * Output:
 *      column
 * Return value:
 *      none
 */
void Butterfly(REGISTER column[8], char pass)
{
  int i;
  REGISTER shadow_column[8];

  /*
   * For readability, we use a shadow column
   * that contains the state of column at the
   * preceding stage of the butterfly.
   */

  /*
   * Initialization
   */

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * First Phase
   */

  Rotate(column+2, column+6, pass-2, pass-1, cpo8,   spo8,   1);
  Rotate(column+1, column+7, pass-1, pass-1, cpo16,  spo16,  1);
  Rotate(column+3, column+5, pass-1, pass-1, c3po16, s3po16, 1);

  if (pass) {
    int a, tmp=column[4], b=column[0];
    a = b+tmp;
    b = b-tmp;
    column[0] = (a - ((tmp<0) ? 1 : 0)) >> 1;
    column[4] = (b - ((tmp<0) ? 1 : 0)) >> 1;
  }
  else {
    column[0] = shadow_column[0] + shadow_column[4];
    column[4] = shadow_column[0] - shadow_column[4];
  }

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Second Phase
   */

  column[1] = shadow_column[1] - shadow_column[3];
  column[3] = shadow_column[1] + shadow_column[3];

  column[7] = shadow_column[7] - shadow_column[5];
  column[5] = shadow_column[7] + shadow_column[5];

  column[0] = shadow_column[0] + shadow_column[6];
  column[6] = shadow_column[0] - shadow_column[6];

  column[4] = shadow_column[4] + shadow_column[2];
```

```
  column[2] = shadow_column[4] - shadow_column[2];

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Third Phase
   */

  column[7] = shadow_column[7] - shadow_column[3];
  column[3] = shadow_column[7] + shadow_column[3];

  column[1] = Multiply(OoR2, shadow_column[1], -2);
  column[5] = Multiply(OoR2, shadow_column[5], -2);

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Fourth Phase
   */

  column[4] = shadow_column[4] + shadow_column[3];
  column[3] = shadow_column[4] - shadow_column[3];

  column[2] = shadow_column[2] + shadow_column[7];
  column[7] = shadow_column[2] - shadow_column[7];

  column[0] = shadow_column[0] + shadow_column[5];
  column[5] = shadow_column[0] - shadow_column[5];

  column[6] = shadow_column[6] + shadow_column[1];
  column[1] = shadow_column[6] - shadow_column[1];

  return;
}

/*
 * IDCT():
 *      Perform 2D IDCT on a block
 * Input:
 *      REGISTER block[64]
 * Output:
 *      block
 * Return value:
 *      none
 */
void IDCT(REGISTER block[64])
{
  int i;

  Scale(block, -4);

  for (i=0; i<8; i++)
    Butterfly(block+8*i, 0);

  Transpose(block);

  for (i=0; i<8; i++)
    Butterfly(block+8*i, 1);

  Round(block, 6, -256, 255);

  Swap(block);
}
```

For informative purposes, a related forward discrete cosine transform (FDCT) implementation is shown below. This fixed-point FDCT does not form an integral part of this Recommendation.

```
/*****************************************************************************
 *
 *                              FIXED-POINT FDCT
 *
 * Fixed-point fast, separable fdct
 * Storage precision: 16 bits signed
```

```
 * Internal calculation precision: 32 bits signed
 * Input range: 9 bits signed, stored in 16 bits
 * Output range: [-2048, +2047]
 * All operations are signed
 *
 ****************************************************************************/

/*
 * Function declarations
 */

void FButterfly(REGISTER column[8]);
void FDCT(REGISTER block[64]);

/*
 * FButterfly():
 *      Perform 1D FDCT on a column
 * Input:
 *      REGISTER column[8]
 * Output:
 *      column
 * Return value:
 *      none
 */
void FButterfly(REGISTER column[8])
{
  int i;
  REGISTER shadow_column[8];

  /*
   * For readability, we use a shadow column
   * that contains the state of column at the
   * preceding stage of the butterfly.
   */

  /*
   * Initialization
   */

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * First Phase
   */

  for (i=0; i<4; i++) {
    column[i]   = shadow_column[i] + shadow_column[7-i];
    column[7-i] = shadow_column[i] - shadow_column[7-i];
  }

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Second Phase
   */

  column[0] = shadow_column[0] + shadow_column[3];
  column[3] = shadow_column[0] - shadow_column[3];

  column[1] = shadow_column[1] + shadow_column[2];
  column[2] = shadow_column[1] - shadow_column[2];

  column[4] = Multiply(OoR2, shadow_column[4], -2);
  column[7] = Multiply(OoR2, shadow_column[7], -2);

  column[6] = shadow_column[6] - shadow_column[5];
  column[5] = shadow_column[6] + shadow_column[5];

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Third Phase
   */
```

```
  column[0] = shadow_column[0] + shadow_column[1];
  column[1] = shadow_column[0] - shadow_column[1];

  column[6] = shadow_column[6] - shadow_column[4];
  column[4] = shadow_column[6] + shadow_column[4];

  column[7] = shadow_column[7] - shadow_column[5];
  column[5] = shadow_column[7] + shadow_column[5];

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Fourth Phase
   */

  Rotate(column+2, column+3, -2, -1, cpo8, spo8, 0);
  Rotate(column+4, column+5, -1, -1, cpo16, spo16, 0);
  Rotate(column+6, column+7, -1, -1, c3po16, s3po16, 0);

  return;
}

/*
 * FDCT():
 *      Perform 2D FDCT on a block
 * Input:
 *      REGISTER block[64]
 * Output:
 *      block
 * Return value:
 *      none
 */
void FDCT(REGISTER block[64])
{
  int i;

  for (i=0; i<8; i++)
    FButterfly(block+8*i);

  Transpose(block);

  for (i=0; i<8; i++)
    FButterfly(block+8*i);

  Round(block, 3, -2048, 2047);

  Swap(block);
}
```

W.6 Picture Message

The picture message function indicates the presence of one or more octets representing message data. The first octet of the message data is a message header with the following structure.

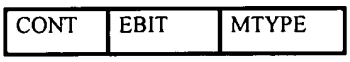


FIGURE W.1/H.263

Structure of first message octet

DSIZE shall be equal to the number of octets in the message data corresponding to a picture message function, including the first octet shown in Figure W.1.

Decoders shall parse picture message data as required by basic PSUPP syntax, but decoder response to picture messages is otherwise undefined.

W.6.1 Continuation (CONT) (1 bit)

If equal to "1", CONT indicates that the message data associated with this picture message function is part of the same logical message as the message data associated with the next picture message function. If equal to "0", CONT indicates that the message data associated with this picture message function terminates the current logical message. CONT may be used, for example, to represent logical messages that span more than 14 octets.

W.6.2 End Bit Position or Track Number (EBIT) (3 bits)

For non-text picture messages, EBIT specifies the number of least significant bits that shall be ignored in the last message octet. In non-text picture messages, if CONT is "1", or if there is only one message octet (i.e. the octet in Figure W.1), EBIT shall equal "0". The number of valid message bits for a non-text picture message function excluding the CONT/EBIT/MTYPE bits is equal to (DSIZE-1)·8 – EBITS. The number of valid message bits for a logical message may be greater due to continuation.

For picture message types containing text information, EBIT shall contain a text track number. The precise meaning of the text track number is not specified herein, but should indicate a particular type (e.g., language) for the text. Track number zero should be considered the default track.

W.6.3 Message Type (MTYPE) (4 bits)

MTYPE indicates the type of message. The defined types are shown in Table W.2.

TABLE W.2/H.263

MTYPE Message Type Values

0	Arbitrary Binary Data
1	Arbitrary Text
2	Copyright Text
3	Caption Text
4	Video Description Text
5	Uniform Resource Identifier Text
6	Current Picture Header Repetition
7	Previous Picture Header Repetition
8	Next Picture Header Repetition, Reliable TR
9	Next Picture Header Repetition, Unreliable TR
10	Top Interlaced Field Indication
11	Bottom Interlaced Field Indication
12	Picture Number
13	Spare Reference Pictures
14..15	Reserved

W.6.3.1 Arbitrary Binary Data

Arbitrary binary data is used to convey any non - ISO/IEC 10646-1 UTF-8 coded binary message. The interpretation of contents of the arbitrary binary data are outside the scope of this Recommendation, but should begin with some identifying pattern (e.g. a four octet identifier code) to aid in distinguishing one type of such data from others.

W.6.3.2 Arbitrary Text

Arbitrary text is used to convey a generic ISO/IEC 10646-1 UTF-8 coded text message. More specific text messages such as copyright information should be represented with other message types (e.g. copyright text) as appropriate.

W.6.3.3 Copyright Text

Copyright text shall be used only to convey intellectual property information regarding the source or the encoded representation in the bitstream. The copyright message shall be coded according to ISO/IEC 10646-1 UTF-8.

W.6.3.4 Caption Text

Caption text shall be used only to convey caption information associated with the current and subsequent pictures of the bitstream. The caption message shall be coded according to ISO/IEC 10646-1 UTF-8. The caption text shall be inserted in the bitstream as if it were to be displayed in a separate text area where new text is appended at the end of previous text and earlier text scrolled away from the point of insertion. The Form Feed (hexadecimal "0x000C") control code shall be used to indicate clearing of the visible text area. The End of Medium (hexadecimal "0x0019") control code shall be used to indicate "caption off" status. However, this Recommendation puts no restriction on how caption text is actually displayed and stored.

W.6.3.5 Video Description Text

Video description text shall be used only to convey descriptive information associated with the information contents of the current bitstream. The video description shall be coded according to ISO/IEC 10646-1 UTF-8. The video description text shall be inserted in the bitstream as if it were to be displayed in a separate text area where new text is appended at the end of previous text and earlier text scrolled away from the point of insertion. The Form Feed (hexadecimal "0x000C") control code shall be used to indicate clearing of the visible text area. The End of Medium (hexadecimal "0x0019") control code shall be used to indicate "description off" status. However, this Recommendation puts no restriction on how video description text is actually displayed and stored.

W.6.3.6 Uniform Resource Identifier (URI) Text

The message consists of a uniform resource identifier (URI), as defined in IETF RFC 2396. The URI shall be coded according to ISO/IEC 10646-1 UTF-8.

W.6.3.7 Current Picture Header Repetition

The picture header from the current picture is repeated in this message. The repeated bits exclude any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4.

W.6.3.8 Previous Picture Header Repetition

The picture header from the previously transmitted picture is repeated in this message. The repeated bits exclude the first two bytes of picture start code (PSC) and any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4.

W.6.3.9 Next Picture Header Repetition, Reliable TR

The picture header from the next picture to be transmitted is repeated in this message. The repeated bits exclude the first two bytes of picture start code (PSC) and any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4.

W.6.3.10 Next Picture Header Repetition, Unreliable TR

The picture header from the next picture to be transmitted is repeated in this message. The repeated bits exclude the first three bytes of picture header and any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4. Any TR or ETR bits in the repeated picture header are not necessarily the same as the corresponding bits in the next picture header.

W.6.3.11 Interlaced Field Indications

In the case of interlaced field indications, the message consists of an indication of interlaced field coding. This indication does not affect the decoding process. However, it indicates that the current picture was not actually scanned as a progressive-scan picture. In other words, it indicates that the current coded picture contains only half of the lines of the full resolution source picture. DSIZE shall be 1, CONT shall be 0, and EBIT shall be 0 for interlaced field indications. In the case of interlaced field coding, each increment of the temporal reference denotes the time between the sampling of alternate half-picture fields of a picture, rather than the time between two complete pictures. In the case of a top interlaced field indication, the current picture contains the first (i.e., top), third, fifth, etc. lines of the complete picture. In the case of a bottom interlaced field indication, the current picture contains the second, fourth, sixth, etc. lines of the complete picture. When sending interlaced field indications, an encoder shall conform to the following conventions:

1. The encoder shall use a picture clock frequency (custom picture clock frequency, if necessary) such that each new field of the original source video corresponds to an increment of 1 in the temporal reference.
2. The encoder shall use a picture size (custom picture size, if necessary) such that the picture dimensions correspond to those of a single field.
3. The encoder shall use a pixel aspect ratio (custom pixel aspect ratio, if necessary) such that the full-height picture aspect ratio corresponds to the picture aspect ratio derived from the pixel aspect ratio of the single field represented by the current encoded picture.

Interlaced field scanning was introduced originally as an analog video compression technique. Although progressive picture scanning is generally regarded as superior for digital compression and display, the use of interlaced field scanning has persisted in many camera and display designs. Interlaced field coding (which can be implemented with lower delay than either interlaced full-picture coding or progressive-scan picture coding at half the interlaced field rate) is therefore supported by the indications herein.

An encoder shall not send interlaced field indications unless the capability of the decoder to receive and properly process such field-based pictures has been established by external means (for example, Recommendation H.245). Failure to establish such a decoder capability may produce a visually annoying small-amplitude vertical shaking behavior in the decoded picture received and displayed by a decoder.

For example, an encoder may use interlaced field coding with application of the Reference Picture Selection mode (specified in Annex N) or the Enhanced Reference Picture Selection mode (specified in Annex U) to allow the addressing of more than one prior field. For "525/60" interlaced field coding for a 4:3 picture aspect ratio with 704 coded luminance samples per line and 240 coded luminance lines per field, the encoder shall use a custom picture size having a picture width of 704 and a picture height of 240, a custom pixel aspect ratio of 5:11, and a custom picture clock frequency specified with a clock conversion code '1' and a clock divisor of 30. For "625/50" interlaced field coding for a 4:3 picture aspect ratio with 704 coded luminance samples per line and 288 coded luminance lines per field, the encoder shall use a custom picture size having a picture width of 704 and a picture height of 288, a custom pixel aspect ratio of 6:11, and a custom picture clock frequency specified with a clock conversion code '0' and a clock divisor of 36.

The vertical sampling positions of the chrominance samples in interlaced field coding of a top field picture are specified as shifted up by 1/4 luminance sample height relative to the field sampling grid in order for these samples to align vertically to the usual position relative to the full-picture sampling grid. The vertical sampling positions of the chrominance samples in interlaced field coding of a bottom field picture are specified as shifted down by 1/4 luminance sample height relative to the field sampling grid in order for these samples to align vertically to the usual position relative to the full-picture sampling grid. The horizontal sampling positions of the chrominance samples are specified as unaffected by the application of interlaced field coding. The vertical sampling positions are shown with their corresponding temporal sampling positions in Figure W.2/H.263.

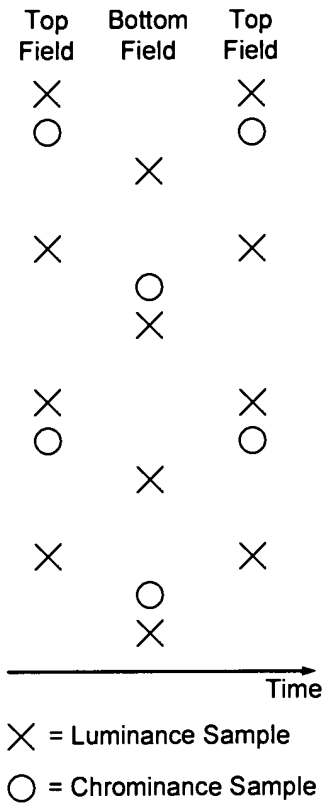


FIGURE W.2/H.263

Vertical and Temporal Alignment of Chrominance Samples for Interlaced Field Coding

W.6.3.12 Picture Number

This message shall not be used if Annex U is in use. The message contains two data bytes that carry a 10-bit Picture Number. Consequently, DSIZE shall be 3, CONT shall be 0, and EBIT shall be 6. Picture Number shall be incremented by 1 for each coded and transmitted I or P picture or PB or Improved PB frame, in a 10-bit modulo operation. For EI and EP pictures, Picture Number shall be incremented for each EI or EP picture within the same scalability enhancement layer. For B pictures, Picture Number shall be incremented relative to the value in the most recent non-B picture in the reference layer of the B picture which precedes the B picture in bitstream order (a picture which is temporally subsequent to the B picture). If adjacent pictures in the same enhancement layer have the same temporal reference, and if the reference picture selection mode (see Annex N) is in use, the decoder shall regard this occurrence as an indication that redundant copies have been sent of approximately the same pictured scene content, and all of these pictures shall share the same Picture Number. If the difference (modulo 1024) of the Picture Numbers of two consecutively received non-B pictures in the same enhancement layer is not 1, and if the pictures do not represent approximately the same pictured scene content as described above, the decoder should infer a loss of pictures or corruption of data.

W.6.3.13 Spare Reference Pictures

Encoders can use this message to instruct decoders which pictures resemble the current motion compensation reference picture so well that one of them can be used as a spare reference picture if the actual reference picture is lost during transmission. If a decoder lacks an actual reference picture but can access a spare reference picture, it should not request for an INTRA picture update. It is up to encoders to choose the spare reference pictures if any. The message data bytes contain the Picture Numbers of the spare reference pictures in preference order (the most preferred appearing first). Picture Numbers refer to the values that are transmitted according to Annex U or section W.6.3.12. This message can be used for P, B, PB, Improved PB, and EP picture types. However, if Annex N or Annex U is in use and if the picture is associated with multiple reference pictures, this message shall not be used. For EP pictures, the message shall be used only for forward prediction, whereas upward prediction is always done from the temporally corresponding reference layer picture. For B, PB, and Improved PB picture types, it specifies a picture for use as a forward motion prediction reference. This message shall not be used if the picture is an I or EI picture.

© ISO/IEC 2002 — All rights reserved

INTERNATIONAL ORGANISATION FOR STANDARDISATION

ORGANISATION INTERNATIONALE DE NORMALISATION

ISO/IEC JTC1/SC29/WG11

CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11

MPEG02/N5303

09 January 2003

Title: **Study Text of ISO/IEC 14496-15/FCD**
Authors: **David Singer, Toby Walker (Editors)**
Group: **MPEG-4**
Subgroup: **Systems**

Document type: International Standard
Document subtype: Amendment
Document stage: (30) Committee
Document language: E

ISO/IEC JTC 1/SC 29 N

Date: 2002-11-08

ISO/IEC 14496-15:2002/StudyCD

ISO/IEC JTC 1/SC 29/WG 11

Secretariat:

Information technology — Coding of audio-visual objects — Part 15: AVC File Format

Élément introductif — Élément central — Partie 1: Titre de la partie

Error! AutoText entry not defined.

Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

[Indicate the full address, telephone number, fax number, telex number, and electronic mail address, as appropriate, of the Copyright Manger of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

© ISO/IEC 2002 — All rights reserved

Contents

Page

INTERNATIONAL ORGANISATION FOR STANDARDISATION 1
Foreword vi
Introduction vii
1 SCOPE 1
2 NORMATIVE REFERENCES 1
3 TERMS AND DEFINITIONS 1
3.1 Terminology 1
3.1.1 Parameter Set 1
3.1.2 Parameter Set Elementary Stream 1
3.1.3 Video Elementary Stream 1
3.2 Symbols and abbreviated terms 2
4 USE OF AVC IN MPEG-4 SYSTEMS 2
4.1 AVC Streams 2
4.1.1 Structure of the AVC streams 2
4.1.2 Time Base for AVC Streams 3
4.1.3 Access unit definition 4
4.1.4 AVC Access Unit Syntax and Semantics 5
4.1.5 Decoder Specific Information 6
4.1.6 AVC DecoderConfigDescriptor 8
4.1.7 Elementary Stream Descriptor 9
4.2 Codepoints 9
4.2.1 visualProfileLevelIndication 9
4.2.2 objectTypeIndication 10
4.3 Extensibility 11
5 File Format 11
5.1 Introduction 11
5.2 Terminology 11
5.3 AVC File type and identification 12
5.4 AVC Track Structure 12
5.5 Canonical order and restrictions 12
5.6 Sample format 13
5.7 Template Fields Used 13
5.8 Visual Width and Height 14
5.9 Sample Table elements 14
5.9.1 Sample description name and format 14
5.9.2 Decoding Time (DTS) and Composition Time (DTS) 15
5.9.3 Sync sample (IDR) 15
5.9.4 Shadow sync 15
5.10 Sample Groups 16
5.10.1 Introduction 16
5.11 Layering and sub-sequences 18
5.12 Alternate Streams and Switching Pictures 20
5.12.1 Alternate Group 21
5.12.2 Track references 22
5.12.3 Sample dependency 22

© ISO/IEC 2002 — All rights reserved

5.13 Representation of new structures in Movie Fragments 23
5.14 Use of the AVC File Format in MPEG-4 Systems 23
5.14.1 Forming the Elementary Stream Descriptor 23
5.14.2 Forming the DecoderConfigDescriptor 23
5.14.3 Profile and Level Indication 23
5.14.4 Elementary Stream Descriptors 23
5.14.5 Switching Picture Tracks 24
5.15 Possible issues 24
5.15.1 Typical patterns of NAL Unit sizes etc 24
5.15.2 Do we need better visibility into where SEIs occur? 24
5.15.3 Do we need to allow parameter set updates in the OD stream? 24
5.15.4 Will chunking be affected by Parameter Set changes? 24
5.15.5 Layering and sub-sequences 24

© ISO/IEC 2002 — All rights reserved

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 14496-15 was prepared by ISO/IEC/TC JTC 1/SC 29.

© ISO/IEC 2002 — All rights reserved

Introduction

The Advanced Video Codec (AVC), jointly developed by the ITU and ISO/IEC SC29/WG11 (MPEG), offers not only increased coding efficiency and enhanced robustness, but also interesting opportunities and challenges for the systems which use it. To enable the best visibility of, and access to, those features, and to enhance the opportunities for the interchange and interoperability of media, this standard defines a storage format for video streams compressed using AVC and using carrying within the MPEG-4 Systems Framework (ISO/IEC 14496-1:2001 and its various amendments).

This specification defines a storage format based on, and compatible with, the ISO Media File format ISO/IEC 14496-12 | 15444-12), which is used by the MP4 file format (ISO/IEC 14496-14) and the Motion JPEG 2000 file format (ISO/IEC 15444-1) among others. This specification enables AVC video streams to:

- be used in conjunction with other media streams, such as audio;
- be used in an MPEG-4 systems environment, if desired;
- be formatted for delivery by a streaming server, using hint tracks;
- inherit all the use cases and features of the ISO Media file format on which MP4 and MJ2 are based.

The ISO Media File format is becoming increasingly common as a general-purpose media container format for the exchange of digital media, and its use in this context should accelerate both adoption and interoperability.

Extensions to the base format are defined here to expose the new systems aspects of the AVC codec.

© ISO/IEC 2002 — All rights reserved

Information technology — Coding of audio-visual objects — Part 15: AVC File Format

1 SCOPE

This document specifies the storage format for AVC (ISO/IEC 14496-10 | ITU Recommendation H.264) video streams. It also specifies how such streams are carried in an MPEG-4 context.

[Ed's Note: It is expected that the specification of carriage of AVC streams in MPEG-4 Systems will move into an appropriate MPEG-4 Systems Amendment].

2 NORMATIVE REFERENCES

[1] ISO/IEC 14496-1:2001, *Information Technology -- Coding of audio-visual objects -- Part 1 :Systems*
[2] ISO/IEC 14496-12, *Information Technology -- Coding of audio-visual objects -- Part 12 :ISO Media File Format*
[3] ISO/IEC 14496-14, *Information Technology -- Coding of audio-visual objects -- Part 14: MP4 File Format*
[4] ISO/IEC 14496-10, *Information technology -- Coding of audio-visual objects -- Part 10: Advanced Video Coding, Final Committee Draft, ISO/SC29/WG11/N4920.*

3 TERMS AND DEFINITIONS

For the purposes of this specification, the terms and definitions specified in normative references ISO/IEC 14496-1 and ISO/IEC 14496-10 | ITU Recommendation H.264 shall apply. In addition, the following terms and abbreviations shall apply to all clauses in this specification.

3.1 Terminology

3.1.1 Parameter Set

A parameter set is either a sequence parameter set or a picture parameter set. This term is used to refer to both types of parameter sets.

3.1.2 Parameter Set Elementary Stream

An elementary stream containing access units made up only of sequence and picture parameter set NAL units synchronized with the video elementary stream.

3.1.3 Video Elementary Stream

An elementary stream containing access units made up of NAL units for coded picture data.

© ISO/IEC 2002 — All rights reserved

3.2 Symbols and abbreviated terms

AVC	ISO/IEC 14496-10, Advanced Video Coding
NAL	Network Abstraction Layer
SEI	Supplementary Enhancement Information

4 USE OF AVC IN MPEG-4 SYSTEMS

4.1 AVC Streams

This subclause specifies the elementary stream format used to carry AVC visual content within the MPEG-4 systems framework.

4.1.1 Structure of the AVC streams

This subclause defines two elementary streams for carriage of AVC content within the MPEG-4 systems framework: a video elementary stream for carrying video data and an optional parameter set elementary stream for carrying parameter sets out-of-band from the video data.

AVC specifies a set of Network Abstraction Layer units, which contain different types of data. While the syntax of each NAL unit is specified by AVC, how NAL units are to be carried as elementary streams in the MPEG-4 Systems framework is not specified by ISO/IEC 14496-10. This subclause specifies the format of MPEG-4 elementary streams for AVC content for use within MPEG-4 Systems.

AVC content shall be carried in MPEG-4 Systems in either one or two elementary streams:

- **Video Elementary Stream**: This elementary stream, structured into video access units, contains video coding related NAL units (i.e. those NAL units containing video data or signaling video structure. If the optional parameter set elementary stream is not present, sequence and picture parameter set NAL units may be present in this stream.

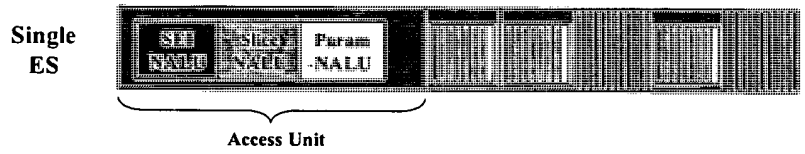
- **Parameter Set Elementary Stream**: This optional stream is used to transmit only sequence and picture parameter set NAL units. If this stream is used, sequence and picture parameter set NAL units shall not be carried in the video elementary stream.

The set types of NAL units are allowed in each of video and parameter set elementary streams is specified in Table 1.

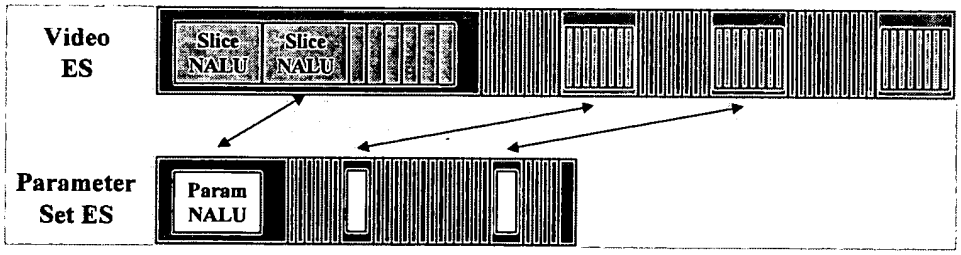
© ISO/IEC 2002 — All rights reserved

Table 1 -- NAL Unit Types in Elementary Streams

Value of nal_unit_type	Description	Video ES	Parameter Set ES
0x0	Reserved for external use	No	No
0x1	Coded slice	Yes	No
0x2	Coded data partition A (DPA)	Yes	No
0x3	Coded data partition B (DPB)	Yes	No
0x4	Coded data partition C (DPC)	Yes	No
0x5	Coded slice of an IDR picture	Yes	No
0x6	Supplemental Enhancement Information (SEI)	Yes	No
0x7	Sequence Parameter Set (SPS)	Yes, if parameter ES is not used.	Yes
0x8	Picture Parameter Set (PPS)	Yes, if parameter ES is not used.	Yes
0x9	Picture Delimiter (PD)	Yes	No
0xA	Filler Data (FD)	Yes	Yes
0xB – 0x17	Reserved	No	No
0x18 – 0x1F	For external use	Not specified by this standard.	Not specified by this standard.



(a) Single Video Elementary Stream containing all NAL Units



(b) Synchronized Video and Parameter with arrows denoting synchronization between streams.

Figure 1: AVC Elementary Stream Structure

4.1.2 Time Base for AVC Streams

The time base associated to an AVC stream shall be indicated using the means provided by MPEG-4 systems. If present, an AVC parameter set elementary stream shall adhere to the same time base as the AVC video elementary stream.

© ISO/IEC 2002 — All rights reserved

4.1.3 Access unit definition

An AVC access unit consists of one or more NAL units, as defined in ISO/IEC 14496-10. All NAL units that are to be processed at the same instant in time shall constitute a single access unit. Access units in AVC streams shall be labelled and time-stamped by suitable means. This shall be done via the related flags and the composition time stamps, respectively, in the SL packet header (see ISO/IEC 14496-1:2001, subclause 10.2.4). The composition time indicates the point in time at which the decoded data shall be composed. The decoding and composition time for an AVC access unit can have different values.

An AVC access unit shall set the `randomAccessPointFlag` in the SL packet header to '1' for this access if ALL of the following conditions are met:

- All video data NAL unit indicate a coded slice of an instantaneous decoder refresh (IDR) picture.
- All sequence and picture parameter sets needed to decode the video data in the access unit are contained in the decoder specific information for the video elementary stream.

Otherwise, the `randomAccessPointFlag` shall be set to '0'.

An AVC access unit shall include the following NAL units:

- **Video Data NAL Units (Coded Slice, Coded DPA, Coded DPB, Coded DPC, Coded IDR)**: All slice and data partition NAL units for a single picture shall be contained with the access unit whose decoding time and composition time is that of the picture. A single picture is one of a frame, a top and bottom field for one frame, or a single top or bottom field. Each AVC access unit shall contain at least one Video Data NAL unit.
- **SEI Message NAL Units**: All SEI NAL units shall be contained in the access unit whose decoding time is that before which the SEI messages come into effect instantaneously. This means that the SEI messages for a picture shall be included in the access unit containing to that picture and that SEI messages pertaining to a sequence of pictures shall be included in the access unit containing the first picture of the sequence to which the SEI message pertains. SEI messages that pertain to the picture level or higher levels shall occur prior to any video data NAL units in the access unit. Slice level messages may occur anywhere within an access unit.
- **Filler Data NAL Units:** Filler data NAL units can be included in any access unit. An access unit shall not consist entirely of filler data NAL units.
- **Picture Delimiter NAL Units**: Picture delimiter NAL units, if present, shall occur once at the beginning of the access unit whose decoding time is that of the picture being delimited.
- **Parameter Sets**: If a single video elementary stream is used to carry AVC content, all sequence and picture parameter sets NAL units shall be contained in the access unit whose decoding time is the time equal or prior to when the parameter sets come into effect. As specified in ISO/IEC 14496-10, any sequence or picture parameter sets present in an access unit shall occur prior to any video coding NAL units and SEI NAL units in that access unit. If a parameter set elementary steam is used, then the access unit in the parameter stream shall have a decoding time equal or prior to when the parameter set(s) comes into effect instantaneously. This means that for a parameter set to be used in a picture it must be sent prior to the access unit containing that picture or in the access unit for that picture.

The following restrictions apply to the Access Units:

© ISO/IEC 2002 — All rights reserved

a) **Timing SEIs**: The Temporal Reference SEI message and the clock timestamp SEI messages that define the timing information shall not be present in the video elementary stream as timing is provided by the MPEG-4 Sync Layer.

b) **Start Codes** shall not be present in the stream.

The sequence of NAL units in an access unit shall be any legal decoding order conforming to that specified in ISO/IEC 14496-10.

4.1.4 AVC Access Unit Syntax and Semantics

This subclause specifies the syntax and semantics of access units for AVC video. Access Units are externally framed and have a size supplied by that external framing. The structure of an AVC access unit is depicted in Figure 2.

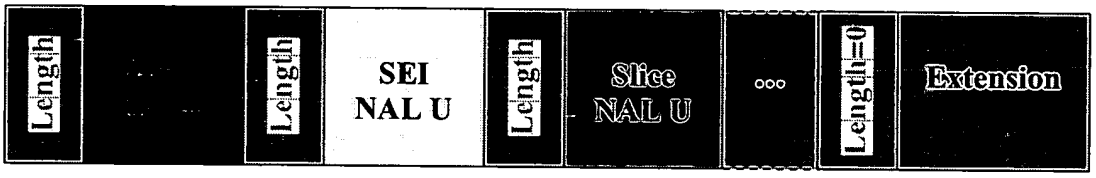


Figure 2: The structure of an AVC Access Unit

An AVC access unit contains a succession of coded picture representations, each of which is made up of a set of NAL units. Each NAL units is is represented with a:

- *Length*: Indicates the length in bytes of the following NAL Unit. The length field can be configured to be of 1, 2, 3 or 4 bytes.
- *NAL Unit*: Contains the NAL unit data as specified in ISO/IEC 14496-10.

If a length field contains the value 0, then all remaining bytes in the access unit contain extension data which is reserved for use in future versions of this specification and must be ignored; those bytes are not NAL Units and must not be passed to the AVC decoder. No coded pictures or their data may occur after such a zero length field. In other words, extension data must be at the very end of the access unit.

4.1.4.1 Syntax

```
class aligned(8) AVCAccessUnit
{
   if(AVCConfigurationRecord.PictureLengthSize > 0)
      unsigned int(8)    PictureCount
   else
      PictureCount = 1;
   for(i=0; i<PictureCount; i++)
   {
      if(AVCConfigurationRecord.PictureLengthSize > 0)
         unsigned int(AVCConfigurationRecord.PictureLengthSize*8)
            PictureLength;
         for (i=0; ; i++;)     // to end of the picture
         {
            unsigned int((AVCConfigurationRecord.LengthSizeMinusOne+1)*8) NALUnitLength;
            if (NALUnitLength==0) {
```

© ISO/IEC 2002 — All rights reserved

```
            bit() reserved; // to end of the picture
            break;
          } else {
            bit(NALUnitLength * 8) NALUnit;
          }
      }
    }
}
```

4.1.4.2 Semantics

An AVC Video Access Unit consists of a set of NAL units with the same decoding and composition time. A single access unit can contain more than one picture if it contains multiple coded representations of the same picture (either redundant pictures or multiple switching pictures). The syntax of an access unit is configured via the decoder specific configuration for the AVC elementary stream. An AVC Parameter Set Access Unit contains one or more sequence and picture parameter set NAL units. If the coded representations of a picture are redundant pictures then the primary picture (redundant_pic_cnt = 0) shall occur first and all subsequence redundant picture shall occur in ascending order of their redundant picture (redundant_pic_cnt) value.

A video elementary stream is a random access point if and only if it contains an IDR picture and either of the following conditions holds:

- All of the sequence and picture parameter sets necessary to decode the IDR picture; or,
- A separate parameter set elementary stream is used.

A parameter set elementary stream access unit is a random access point if and only if it contains all sequence and picture parameter sets used from that time forward in the video coding stream and parameter set stream until the next time any parameter set is updated.

`PictureCount` indicates the number of coded representations of the same picture contained within this unit. Such pictures may not be identical objective representations of the same picture. This field shall have a value of at least one.

`PictureLength` indicates the number of bytes contained the coded data for this picture's representation, which is made up of the set of NAL units preceded by length fields. The length of a picture comprises the number of bytes in all NAL units and the size of all `NALUnitLength` fields. If an access unit includes extension data then the picture length field does not include the size of the data or the `NALUnitLength` field whose value is zero.

`NALUnitLength` indicates the size of a NAL unit measured in bytes. The length field includes the size of both the one byte NAL header and the EBSP payload but does not include the length field itself.

`NALUnit` contains a single NAL unit. The syntax of an NAL unit is defined in ISO/IEC 14496-10 and includes both the one byte NAL header and the variable length encapsulated byte stream payload.

4.1.5 Decoder Specific Information

This subclause specifies the decoder specification information for ISO/IEC 14496-10 video content.

4.1.5.1 AVC Decoder Configuration Record

This record contains the size of the length field used in each access unit to indicate the length of its contained NAL units as well as the initial parameter sets. This record is externally framed (its size must be supplied by the structure which contains it).

© ISO/IEC 2002 — All rights reserved

This record contains a version field. This version of the specification defines version 1 of this record. Incompatible changes to the record will be indicated by a change of version number. Readers must not attempt to decode this record or the streams to which it applies if the version number is unrecognized.

Compatible extensions to this record will extend it and will not change the configuration version code. Readers should be prepared to ignore unrecognized data beyond the definition of the data they understand (e.g. after the parameter sets in this specification).

When used to provide the configuration of a parameter set elementary stream, the configuration record shall contain no sequence or picture parameter sets (`numOfParameterSets` shall have the value 0).

4.1.5.1.1 Syntax

```
aligned(8) class AVCConfigurationRecord {
   unsigned int(8) configurationVersion = 1;
   unsigned int(8) AVCProfileIndication;
   unsigned int(8) AVCLevelIndication;
   unsigned int(8) PictureLengthSize;
   unsigned int(16) nonStoredDegredationPriorityLow;
   unsigned int(16) nonStoredDegredationPriorityHigh;
   bit(6) reserved = '111111'b;
   unsigned int(2) lengthSizeMinusOne;
   unsigned int(8) numOfSequenceParameterSets;
   for (i=0; i< numOfSequenceParameterSets;  i++) {
      unsigned int(16) sequenceParameterSetLength ;
      bit(8*sequenceParameterSetLength) sequenceParameterSetNALUnit;
   }
   unsigned int(8) numOfPictureParameterSets;
   for (i=0; i< numOfPictureParameterSets;  i++) {
      unsigned int(16) pictureParameterSetLength;
      bit(8*pictureParameterSetLength) pictureParameterSetNALUnit;
   }
}
```

4.1.5.1.2 Semantics

`AVCProfileLevelIndication` contains the profile code defined in Annex A of the AVC specification [ISO/IEC 14496-10 | ITU Recommendation H.264].

`AVCLevelIndication` contains the level code defined in Annex A of the AVC specification [ISO/IEC 14496-10 | ITU Recommendation H.264].

`PictureLengthSize` indicates the length of the picture size field used to indicate the size of a coded picture representation. A value of zero indicates that only a single coded picture representation is allowed in an access unit.

`nonStoredDegrediationPriorityHigh` indicates the largest value of the range of degradation priorities that correspond to AVC non-stored frames (nal_type_idc=0). If the value of `nonStoredDegrediationPriorityHigh` exceeds `nonStoredDegrediationPriorityLow` then no correspondence between degradation priorities and non-stored frames is defined.

`nonStoredDegrediationPriorityLow` indicates the smallest value of the range of degradation priorities that correspond to AVC non-stored frames (nal_type_idc=0).

`lengthSizeMinusOne` indicates the length in bytes of the `NALUnitLength` field in an AVC video access unit of the associated stream minus one. For example, a size of one byte is indicated with a value of 0.

`numOfSequenceParameterSets` indicates the number of sequence parameter sets that are used as the initial set of sequence parameter sets for decoding the AVC elementary stream.

© ISO/IEC 2002 — All rights reserved

`sequenceParameterSetLength` indicates the length of the parameter set NAL unit as defined in ISO/IEC 14496-10.

`sequenceParameterSetNALUnit` contains a sequence parameter set NAL Unit, as specified in ISO/IEC 14496-10. Sequence parameter sets shall occur in order of ascending parameter set identifier with gaps being allowed.

`numOfPictureParameterSets` indicates the number of picture parameter sets that are used as the initial set of picture parameter sets for decoding the AVC elementary stream.

`pictureParameterSetLength` indicates the length of the parameter set NAL unit as defined in ISO/IEC 14496-10.

`pictureParameterSetNALUnit` contains a parameter set NAL Unit, as specified in ISO/IEC 14496-10. Picture parameter sets shall occur in order of ascending parameter set identifier with gaps being allowed.

4.1.5.2 AVC Stream DecoderSpecificInfo

This subclause defines the `DecoderSpecificInfo` descriptor for an AVC elementary stream.

4.1.5.2.1 Syntax

```
aligned(8) class AVCDecoderSpecificInfo extends DecoderSpecificInfo : bit(8)
tag=DecSpecificInfoTag {
     AVCVideoConfigurationRecord config;
}
```

4.1.5.2.2 Semantics

The decoder specific information for an AVC stream contains an AVC video stream decoder configuration record.

`config` contains the decoder configuration record for the AVC elementary stream decoder configuration.

4.1.5.3 Parameter Set

Initial sequence and picture parameter sets are supplied in the `DecoderSpecificInfo` for the AVC video stream, as defined in subclause 4.1.5.1. If a parameter set must be updated, then one of two methods can be used:

a) Sequence or Picture Parameter Set NAL units may be inserted in the video stream;

b) A separate parameter set stream may be used to carry parameter sets, possibly using a transport with different error resilience or reliability characteristics.

4.1.6 AVC DecoderConfigDescriptor

The `DecoderConfigDescriptor` shall set the value of `streamtype` equal to 0x04 for both AVC video elementary stream and the AVC parameter set elementary streams.

The `DecoderConfigDescriptor` for an AVC video elementary stream, possibly including in-line sequence or picture parameter sets, shall set `objectTypeIndication` to be 0x21 (ISO/IEC 14496-10 Visual). For the AVC parameter set elementary stream, the `objectTypeIndication` value shall equal 0x22 (ISO/IEC 14496-10 Visual Parameter Sets).

© ISO/IEC 2002 — All rights reserved

4.1.7 Elementary Stream Descriptor

If the parameter set elementary stream is present, the elementary stream descriptors for the two streams shall satisfy the following conditions:

(1) The video elementary stream is dependent on the parameter set elementary stream and the `ES_Descriptor` for the video elementary stream shall have a `streamDependenceFlag` equal to true and indicate the ESID of the parameter in the `dependsOnESID` field. The `streamDependenceFlag` in the `ES_Descriptor` for the parameter set elementary stream shall be false.

(2) The elementary stream clocks for the parameter set elementary stream and the video elementary stream shall be the same and synchronized. The `OCRstreamflag` and `OCR_ES_Id` fields in the `ESDescriptor` for the video elementary stream and parameter set elementary streams shall be used to indicate that both streams share the same OCR.

4.2 Codepoints

4.2.1 visualProfileLevelIndication

Ed's Note: What should the visual profile level indication in the initial object descriptor be? Is it even necessary to add a new one? The list below does not seem up to date anyway and I do not recall any amendments to MPEG-4 Systems that touch this table.

Replace Table 6 in ISO/IEC 14496 with the following table:

Table 2 - visualProfileLevelIndication Values

Value	Profile	Level
0x00	Reserved for ISO use	-
0x01	Simple	L3
0x02	Simple	L2
0x03	Simple	L1
0x04	Simple Scalable	L2
0x05	Simple Scalable	L1
0x06	Core	L2
0x07	Core	L1
0x08	Main	L4
0x09	Main	L3
0x0A	Main	L2
0x0B	N-Bit	L2
0x0C	Hybrid	L2
0x0D	Hybrid	L1
0x0E	Basic Animated Texture	L2
0x0F	Basic Animated Texture	L1
0x10	Scalable Texture	L3
0x11	Scalable Texture	L2
0x12	Scalable Texture	L1
0x13	Simple Face Animation	L2
0x14	Simple Face Animation	L1

© ISO/IEC 2002 — All rights reserved

0x15	Advanced Visual Coding	-
0x15-0x7F	reserved for ISO use	-
0x80-0xFD	user private	-
0xFE	no visual profile specified	-

4.2.2 objectTypeIndication

Replace Table 8 in ISO/IEC 14496 with the following table:

Table 3 - objectTypeIndication Values

Value	`ObjectTypeIndication` Description
0x00	Forbidden
0x01	Systems ISO/IEC 14496-1 [a]
0x02	Systems ISO/IEC 14496-1 [b]
0x03-0x1F	reserved for ISO use
0x20	Visual ISO/IEC 14496-2 [c]
0x21	Visual ISO/IEC 14496-10 [e]
0x22	Visual ISO/IEC 14496-10 [e]
0x23-0x3F	reserved for ISO use
0x40	Audio ISO/IEC 14496-3 [d]
0x41-0x5F	reserved for ISO use
0x60	Visual ISO/IEC 13818-2 Simple Profile
0x61	Visual ISO/IEC 13818-2 Main Profile
0x62	Visual ISO/IEC 13818-2 SNR Profile
0x63	Visual ISO/IEC 13818-2 Spatial Profile
0x64	Visual ISO/IEC 13818-2 High Profile
0x65	Visual ISO/IEC 13818-2 422 Profile
0x66	Audio ISO/IEC 13818-7 Main Profile
0x67	Audio ISO/IEC 13818-7 LowComplexity Profile
0x68	Audio ISO/IEC 13818-7 Scaleable Sampling Rate Profile
0x69	Audio ISO/IEC 13818-3
0x6A	Visual ISO/IEC 11172-2
0x6B	Audio ISO/IEC 11172-3
0x6C	Visual ISO/IEC 10918-1
0x6D - 0xBF	reserved for ISO use
0xC0 - 0xFE	user private
0xFF	no object type specified

© ISO/IEC 2002 — All rights reserved

[a] This object type shall be used for all streamTypes defined in ISO/IEC 144961 except IPMP streams.

[b] Includes associated Amendment(s) and Corrigendum(a)

[c] Includes associated Amendment(s) and Corrigendum(a). The actual object types are defined in ISO/IEC 14496-2 and are conveyed in the DecoderSpecificInfo as specified in ISO/IEC 144962, Annex K.

[d] Includes associated Amendment(s) and Corrigendum(a). The actual object types are defined in ISO/IEC 14496-3 and are conveyed in the DecodeSpecificInfo as specified in ISO/IEC 144963 subpart 1 subclause 6.2.1.

[e] Includes associated Amendment(s) and Corrigendum(a). The actual object types are defined in ISO/IEC 14496-10 and are conveyed in the DecoderSpecificInfo as specified in this admendmat, subclause 4.1.5.

4.3 Extensibility

The configuration description format may be extended in future versions of this specification. Similarly, the definition of an AVC Access Unit allows for extension by future versions of this specification.

5 File Format

5.1 Introduction

This subclause defines how to store ISO/IEC 14496-10 data in the MP4 file format. The storage of AVC content uses the existing capabilities of the MP4 file format but also defines new features to support the following features of the AVC codec in the MP4 file format:

- **Switching Pictures**: To enable switching between different coded streams and substitution of pictures within the same stream.
- **Enhanced Group of Pictures**: Provides a structuring of the dependencies of a group of pictures to provide both random access and flexible stream structure (e.g. temporal scalability and layering).
- **Parameter Sets**: The values needed to decode a video coding data are grouped together into a single picture parameter set, which is referenced from each slice.

5.2 Terminology

Table 4 summarizes the correspondences between the sets of terminology used in AVC Video, MPEG-4 Systems (ISO/IEC 14494-1), and MP4 (ISO/IEC 14496-1

Table 4: Correspondence of Terms in AVC, MPEG-4 Systems, and MP4

AVC	MPEG-4 Systems	MP4
-	Presentation	Movie
Stream	Elementary Stream	Track
Picture	Access Unit	Sample

© ISO/IEC 2002 — All rights reserved

5.3 AVC File type and identification

Conformance with this specification is indicated by the presence of the brand 'jvt1' in the compatible-brands list of the `FileTypeBox` as defined in the ISO Media File Format specification [ISO 14496-1:2001, Amd. 5].

Readers conformant to this specification should read the file if this brand occurs in the compatible-brands list. Other structures and/or track types, defined in other specifications, may be present, and these may be ignored by a conformant reader.

If the major-brand is 'jvt1', then

a) the minor-version, as defined in this version of this specification, shall be 1.

b) the file extension shall be ".jvt" [to be checked for collision]

c) the mime type used to serve the file shall be "video/jvt" [to be allocated]

5.4 AVC Track Structure

In the terminology of the ISO Media File Format Specification, an AVC track is a video or visual track. It therefore uses:

a) a `handler_type` of `'vide'` in the `HandlerBox`;

b) a video media header `'vmhd'`;

c) and, as defined below, a derivative of the `VisualSampleEntry`.

5.5 Canonical order and restrictions

The AVC elementary stream is stored in the MP4 file format in a *canonical* format. The canonical format is as *neutral* as possible so that systems that need to customize the stream for delivery over different transport protocols – MPEG-2 Systems, RTP, and so on — should not have to *remove* information from the stream while being free to *add* to the stream. Furthermore, a canonical format allows such operations to be performed against a known initial state.

The canonical stream format is an AVC video stream that satisfies the following conditions:

- *No parameter sets in the video elementary stream.* All sequence and picture parameter set information shall be stored in the sample descriptions for the track or in a separate parameter set track containing a parameter set elementary stream. This ensures that it is not necessary to examine every part of the elementary stream to find relevant parameter sets. It also avoids dependencies of indefinite duration between the sample that contains the parameter set definition and the samples that use it. Parameter sets may be inserted into the video elementary stream when the file is streamed over a transport that permits such parameter set updates.

- *Parameter Set Track:* If present, a parameter set track shall be constructed so that all parameter set samples are random access points. This implies that each there shall be a parameter set access unit at each point a parameter set is updated. Each parameter set shall contain exactly the sequence and picture parameter sets needed to decode

Note: The use of a parameter track in the file format does not require that a system delivering AVC content a use a separate elementary stream for parameter sets. Instead, implementations may choose to map parameter sets to

© ISO/IEC 2002 — All rights reserved

in-band parameter set NAL units in the video elementary stream or use some out-of-band delivery mechanism defined by the transport layer.

- *All timing information is external to stream.* Timing SEI messages that define presentation or composition timestamps shall not be included in the AVC elementary stream. All timing information is stored in the various sample metadata tables. If timing information were embedded in the stream, then there would be a great danger of inconsistency, and a key feature of MP4, that editing can take place without modifying elementary streams, would break.

- *All SEI messages are at the beginning of an AU.* NAL units included in an access unit shall be ordered so that all SEI messages occur before all video NAL units (slices or data partitions) in the access unit. This allows a rapid inspection of the sample to determine whether it contains SEI messages. All SEI messages shall be contained in a single SEI message NAL unit.

- *Redundant Picture*: NAL units within a single access shall be ordered in non-decreasing order of redundant picture count (redundant_pic_cnt).

- *Slice Groups*: NAL units within a single picture (the same value of picture order count) shall be ordered in non-decreasing order of slice group identifier. Within the same slice group, slices shall be ordered by their first Macroblock location (first_mb in the slice header).

- *Partition Ordering:* All data partitions of a slice shall occur in the order: 'A' partition, 'B' partition, and 'C' partition. No interleaving of partitions from different slices is allowed.

- *No start codes.* The elementary stream shall not include start codes. As stored, each NAL Unit is preceded by a length field; this enables easy scanning of the sample's NAL Units. Systems that use start-codes will reformat the stream to use start codes.

- *Picture delimiter NAL units.* Picture delimiter NAL units may be present in the sample data. If present, they shall occur as the first NAL unit in a sample.

- *No filler data.* Video data is naturally represented as VBR in the file format and should be filled for transmission if needed. Filler NAL units and Filler SEI messages shall not be present in the file format stored stream.

Note that these restrictions are more restrictive than the use of AVC in MPEG-4 systems.

5.6 Sample format

A file format sample is a picture in AVC terms, and an access unit in MPEG-4 terms. The sample data shall follow the syntax and semantics defined for AVC video access units in subclause 4.1.4. Note, this format is used for storing data only and does not require the use of MPEG-4 systems framework to carry AVC visual data.

5.7 Template Fields Used

The ISO Media File Format defines a number of fields which have default values but which may be defined for use by specific sub-systems. Tracks containing AVC data use the following template fields:

a) `alternate_group` in the `TrackHeaderBox` (see subclause 5.12 on stream switching);

© ISO/IEC 2002 — All rights reserved

5.8 Visual Width and Height

The width and height fields in a `VisualSampleEntry` must correctly document the encoded width and height as indicated by the sequence parameter sets in the AVC stream that is described by that entry. These *must* correspond to the picture size established by the vui_pic_parameters, which includes optional frame cropping information. The width and height fields do *not* reflect any changes in size caused by SEI messages such as Pan-scan. The visual handling of SEI messages such as pan-scan is both optional and terminal-dependent. If the width and height of the sequence changes, then a new sample description is needed.

The width and height fields in the track header may not be the same as the width and height fields in the one or more `VisualSampleEntry` in the video track. As specified in the ISO Media File Format, if normalized visual presentation is needed, all the sequences are normalized to the track width and height for presentation.

5.9 Sample Table elements

5.9.1 Sample description name and format

5.9.1.1 Sample Description Box

5.9.1.1.1 Definition

Box Types: 'jvt1', 'jvtC', 'm4ds'
Container: Sample Table Box ('stbl')
Mandatory: Yes
Quantity: Exactly one

An AVC visual sample entry shall contain an AVC Visual Stream Configuration Box, as defined below. This includes a `AVCVideoDecoderConfigurationRecord`, as defined in subclause 4.1.5.1 as the content of `DecoderSpecificInfo` for MPEG-4.

Extension descriptors that should be inserted into the Elementary Stream Descriptor, when used in MPEG-4, may also be present.

Multiple sample descriptions may be used, as permitted by the ISO Media File format specification, to indicate sections of video that use different configurations of parameter sets.

5.9.1.1.2 Syntax

```
   // Visual Sequences
class AVCConfigurationBox extends Box('jvtC') {
   AVCVideoConfigurationRecord() AVCConfig;
}

class MPEG4DescriptorsBox extends Box('m4ds') {
   Descriptor Descr[0 .. 255];
}

class AVCSampleEntry() extends VisualSampleEntry ('jvt1'){
   AVCConfigurationBox  config;
   MPEG4ExtensionDescriptorsBox (); // optional
}
```


© ISO/IEC 2002 — All rights reserved

5.9.1.1.3 Semantics

`Compressorname` in the base class `VisualSampleEntry` indicates the name of the compressor used with the value `"\016JVT/AVC Coding"` being recommended (\016 is 14, the length of the string as a byte)

`config` is defined in subclause 4.1.5.2.

`Descr` is a descriptor which should be placed in the `ElementaryStreamDescriptor` when this stream is used in an MPEG-4 context. This does not include `SLConfigDescriptor` or `DecoderConfigDescriptor`, but includes the other descriptors in order to be placed after the `SLConfigDescriptor`. The use of this field is specified in subclause 5.14.1.

5.9.1.2 Parameter Sets

Each AVC sample description, which contains the AVC video stream decoder specific information, includes a group of sequence and picture parameter sets. This group of parameter sets functions much like a codebook. Each parameter set has an identifier, and each slice references the parameter set it was coded against using the parameter set's identifier.

In the file format each configuration of parameter sets is represented separately. A parameter set cannot be updated without causing a different sample description to be used. For example, suppose that samples 1-4 use picture parameter sets identified as 1, 2, 3 and a single sequence parameter set identified as 1. At sample 5 a new value of picture parameter set 2 is required but picture parameter sets 1 and 3 remain unaltered and are used until sample 10. In this case, the sample description for samples 1 through 4 is the same and contains the initial values of picture parameter sets 1, 2, 3 and sequence parameter set 1. At sample 5 the sample description must change to a second sample description, which contains the updated value for picture parameter set 2 *as well as* the original values of picture parameter sets 1 and 3 and sequence parameter set 1. This second sample description is used for samples 5 through 10.

Systems wishing to send sequence or picture parameter set updates will need to compare the two configurations to find the differences in order to send the appropriate parameter set updates.

5.9.2 Decoding Time (DTS) and Composition Time (DTS)

Samples are stored in the file format in decoding order. If frame reordering is not used, this is also presentation order, and only the time-to-sample 'stts' table is used. Note that any kind of picture may be reordered in AVC video, not only B-frames.

If decoding time must be stored, the composition time-to-sample 'ctts' is also used.

5.9.3 Sync sample (IDR)

A sync sample shall correspond to an instantaneously decodable picture in the AVC stream (also known as an Instantaneously Decodable Refresh point (IDR)).

5.9.4 Shadow sync

The use of the shadow sync table to indicate alternate encodings of a sample for random access are supported for as defined in the ISO Base Media File format. A shadow sync shall indicate a sample that is an independently decodable sample, which shall be encoded as either an I-frame or a SI-frame.

While the use of shadow sync for AVC content is supported for backward compatibility reasons, this use is deprecated and use of the mechanisms defined in subclause 5.12 is recommended.

© ISO/IEC 2002 — All rights reserved

5.10 Sample Groups

5.10.1 Introduction

This subclause specifies a generic mechanism for representing a partition of the samples in a track. A *sample grouping* is an assignment of each sample in a track to be a member of one *sample group,* based on a grouping criterion. A sample group in a sample grouping is not limited to being contiguous samples and may contain non-adjacent samples. As there may be more than one sample grouping for the samples in track, each sample grouping has a type field to indicate the type of grouping. For example, a file might contain two sample groupings for the same track: one based on an assignment of sample to layers and another to sub-sequences.

Sample groupings are represented by two linked data structures: (1) a `SampleToGroup` box represents the assignment of samples to sample groups; (2) a `SampleGroupDescription` box contains a *sample group entry* for each sample group describing the properties of the group. There may be multiple instances of the `SampleToGroup` and `SampleGroupDescription` boxes based on different grouping criteria. These are distinguished by a type field used to indicate the type of grouping.

One example of using these tables is to represent the assignments of samples to *layers.* In this case each sample group represents one layer, with an instance of the `SampleToGroup` box describing which layer a sample belongs to. For more details, please refer to subclause 5.11

5.10.1.1 SampleToGroup Box

5.10.1.1.1 Definition

Box Type: 'sbgp'
Container: Sample Table Box ('stbl')
Mandatory: No
Quantity: Zero or more.

This table can be used to find the group that a sample belongs to and the associated description of that sample group. The table is compactly coded with each entry giving the index of the first sample of a run of samples with the same sample group descriptor. The sample group description ID is an index that refers to a `SampleGroupDescription` box, which contains entries describing the characteristics of each sample group.

There may be mutiple instance of this box if there is more the one sample grouping for the samples in a track. Each instance of the `SampleToGroup` box has a type code that distinguishes different sample groupings. Within a track, there shall be a most instance of this box with a particular grouping type. The associated `SampleGroupDescription` shall indicate the same value for the grouping type.

5.10.1.1.2 Syntax

```
aligned(8) class SampleToGroupBox
   extends FullBox('sbst', version = 0, 0)
{
   unsigned int(32)  grouping_type;
   unsigned int(32)  entry_count;
   for (i=1; i <= entry_count; i++)
```


© ISO/IEC 2002 — All rights reserved

```
    {
        unsigned int(32)  sample_count;
        unsigned int(32)  group_description_index;
    }
}
```

5.10.1.1.3 Semantics

`version` is an integer that specifies the version of this box.

`grouping_type` is an integer that identifies the type (i.e. criterion used to form the sample groups) of the sample grouping and links it to its sample group description table with the same value for grouping type. At most once occurrence of this box with the same value for `grouping_type` value shall exist for a track.

`entry_count` is an integer that gives the number of entries in the following table.

`sample_count` is an integer that gives the number of consecutive samples in the group.

`group_description_index` is an integer that gives the index of the sample group entry which describes the samples in this group. The index ranges from 1 to the number of sample group entries in the `SampleGroupDescription` Box.

5.10.1.2 Sample Group Description Box

5.10.1.2.1 Definition

Box Types: 'sgpd'
Container: Sample Table Box ('stbl')
Mandatory: No
Quantity: Zero or more, with one for each Sample Group Table box.

This description table gives information about the characteristics of sample groups. The descriptive information is any other information needed to define or characterize the sample group.

There may be mutiple instance of this box if there is more the one sample grouping for the samples in a track. Each instance of the `SampleGroupDescription` box has a type code that distinguishes different sample groupings. Within a track, there shall be a most instance of this box with a particular grouping type. The associated `SampleToGroup` shall indicate the same value for the grouping type.

The information is stored in the sample group description box after the entry-count. An abstract entry type is defined and sample groupings shall define derived types to represent the description of each sample group. For video tracks, an abstract `VisualSequenceEntry` is used with similar types for audio and hint tracks.

5.10.1.2.2 Syntax

```
// Sequence Entry
abstract class SampleGroupDesriptionEntry (unsigned int(32) handler_type)
{
}

// Visual Sequence
abstract class VisualSampleGroupEntry (type) extends SampleGroupDesriptionEntry
(type)
```

© ISO/IEC 2002 — All rights reserved

```
{
}

// Audio Sequences
abstract class AudioSampleGroupEntry (type) extends SampleGroupDesriptionEntry(type)
{
}

aligned(8) class SampleGroupDescriptionBox (unsigned int(32) handler_type)
   extends FullBox('sgpd', 0, 0){
   unsigned int(32) grouping_type;
   unsigned int(32) entry_count;
   int i;
   for (i = 1 ; i <= entry_count ; i++){
      switch (handler_type){
         case 'vide': // for video tracks
            VisualSampleGroupEntry ();
            break;
         case 'soun': // for audio tracks
            AudioSampleGroupEntry();
            break;
         case 'hint': // for hint tracks
            HintSampleGroupEntry();
            break;
      }
   }
}
```

5.10.1.2.3 Semantics

`version` is an integer that specifies the version of this box.

`grouping_type` is an integer that identifies the `SampleToGroup` box that is associated with this sample group description.

`entry_count` is an integer that gives the number of entries in the following table.

5.11 Layering and sub-sequences

Streams may be constructed so that the referential dependencies between samples allow only subsets of the samples to be sent to the decoder. This mechanism is called *thinning* a stream and is entirely transparent to the decoder receiving the elementary stream. Thinning discards entire sets of samples using knowledge of what other sets of pictures this set of pictures depends on and what picture sets in turn depend on it.

The referential dependencies between samples in a stream are structured into *layers* and *sub-sequences*. Samples in higher layers can only depend on samples in lower layers. Layers are numbered, and the samples organized such that a sample in layer N has no dependencies on samples in layers greater than N.

© ISO/IEC 2002 — All rights reserved

Sub-sequences are sets of samples that have external referential dependencies on other sets of samples. Dependency relations between sub-sequences represent the dependency structure of a stream. The samples in a sub-sequence can be disposed of as a unit when thinning streams. Each sub-sequence belongs to one and only one layer. A sample shall reside in one layer and in one sub-sequence only.

Layering and sub-sequences are not features of the codec but this information is represented in the file format to allow systems reading the files to understand the ways in which stream thinning may be achieved without having to examine the dependency structure of every sample.

5.11.1.1 Sub-Sequence Description Entry

Box Types: 'jvss"
Container: Sample Group Description Box ('sgpd')
Mandatory: No
Quantity: Zero or more.

A sub-sequence description entry is a sample group entry that describes a sub-sequence. A sub-sequence is a set of samples in a track belonging to the same layer. A sub-sequence depends on another sub-sequence if and only if there exists a sample in the this sub-sequence that is directly referentially dependent on some sample in the other sub-sequence. All sample in a sub-sequence shall be depend only on other samples within the same sub-sequence or samples in the sub-sequences on which is it dependent. A sub-sequence can depend on zero or more sub-sequences in the same or lower layers. A sub-sequence shall not depend on sub-sequences in the higher scalability layers.

5.11.1.1.1 Syntax

```
aligned(8) class DependencyInfo
{
     unsigned int(8)   layerNumber;
     unsigned int(32)  subSequenceIdentifier;
}

class AVCSubSequenceEntry () extends VisualSampleGroupEntry ('jvss')
{
     unsigned int(32) subSequenceIdentifer;
     unsigned int(32) avgBitRate;
     unsigned int(32) avgFrameRate;
     unsigned int(16) numReferences;
     DependencyInfo dependency[numReferences];
   }
}
```

5.11.1.1.2 Semantics

`layerNumber` and `subSequenceIdentifier` within the `DependencyInfo` class identify a sub-sequence that is used as a reference for this sub-sequence. Only direct, not indirect, referential dependencies shall be identified.

`subSequenceIdentifier` in the Sub-Sequence Box gives the identifier for the sub-sequence.

`avgBitRate` gives the average bit rate in bits/second of the sub-sequence within the segment. All NAL units belonging to pictures in the layer are taken into account in the calculation. A value of zero means an undefined bit rate.

`avgFrameRate` gives the average frame rate in frames/(256 seconds) of the sub-sequence within the segment. A value of zero means an undefined frame rate.

© ISO/IEC 2002 — All rights reserved

`numReferences` gives the number of sub-sequences directly referenced in this sub-sequence. `dependency` is an array of `DependencyInfo` structures giving the identification information of the referenced sub-sequences.

5.11.1.2 Layer Description Entry

Box Types: 'jvll'
Container: Sample Group Description Box ('sgpd')
Mandatory: No
Quantity: Zero or more.

A layer sample group entry defines the layer information for all samples in a layer. Layers are ordered hierarchically based on their dependency on each other: The first or base layer is independently decodable. The second and subsequent layers are enhancement layers that depend on some data in the base layer. Pictures in a higher layer can only depend on pictures in lower layers.

5.11.1.2.1 Syntax

```
class AVCLayerEntry() extends VisualSampleGroupEntry ('jvll')
{
   unsigned int(8)  layerNumber;
   unsigned int(32) avgBitRate;
   unsigned int(32) avgFrameRate;
}
```

5.11.1.2.2 Semantics

`layerNumber` gives the number of this layer with the base layer being numbered as zero and all enhancement layers being numbered as one or higher with consecutive numbers.

`avgBitRate` gives the average bit rate in bits/second of the layer within the segment. A value of zero indicates an undefined bit rate.

`avgFrameRate` gives the average frame rate in frames per 1000 seconds of this layer with a value of zero indicating an undefined frame rate.

5.12 Alternate Streams and Switching Pictures

In typical streaming scenarios, one of the key requirements is to scale the bit rate of the compressed data in response to changing network conditions. The simplest way to achieve this is to encode multiple streams with different bandwidths and quality settings for representative network conditions. The server can then switch amongst these pre-coded streams in response to network conditions. In current standards, switching between streams is only possible at I-frames, because the frames can only be switched when there are no dependencies on prior pictures for reconstruction.

AVC has adopted SP-pictures and SI-pictures ("switching pictures") that allow switching from one stream to another while still supporting inter-frame coding of switching pictures. Figure 3 shows how SP frames are used to switch between two different bit streams.

© ISO/IEC 2002 — All rights reserved

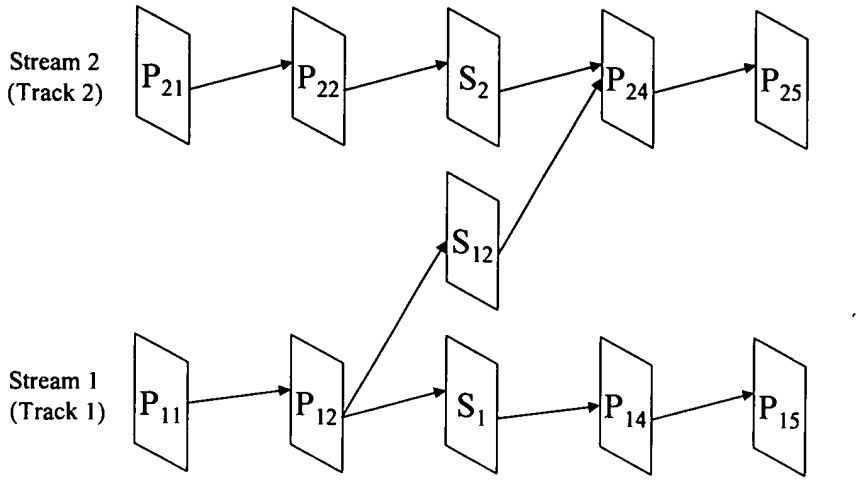


Figure 3: Stream Switching

In the file format, switching pictures are stored in *switching picture* tracks, which are tracks separate from the track that is being switched from and the track being switched to. Switching picture tracks can be identified by the existence of a specific required track reference in that track. A switching picture is an alternative to the sample in the destination track that has exactly the same decoding time. If all switching pictures are SI pictures, then no further information is needed.

If any of the pictures in the switching track are SP pictures, then two extra pieces of information may be needed. First, the source track, that is being switched from, must be identified by using a track reference (the source track may be the same track as the destination track). Second, the dependency of the switching picture on the samples in the source track may be needed, so that a switching picture is only used when the pictures on which it depends have been supplied to the decoder.

This dependency is represented by means of an optional extra sample table. There is one entry per sample in the switching track. Each entry records the relative sample numbers in the source track on which the switching picture depends. If this array is empty for a given sample, then that switching sample contains an SI picture. If the dependency box is not present, then only SI-frames shall be present in the track.

A switching sample may have multiple coded representations with different dependencies. For AVC video, the multiple representations are stored in the same sample (i.e. access unit) using the existing facility to have multiple coded representations in the same access unit. For example, one switch sample might contain a SP-picture representation dependent on some earlier samples, used for stream switching, and another representation as an SI-picture, used for random access.

5.12.1 Alternate Group

The ISO media file format (but not the version one specification of the MPEG-4 file format, which is branded as 'mp41') supports the use of what is called *alternate tracks.* Each track can optionally specify an *alternate group* (in the track header box) that groups together alternate encodings of the same content. Thus, each alternate bit-stream can be stored as a separate track in the MP4 and related together as alternate tracks. All the tracks which form a group which may be switched between, but not the tracks containing the switching pictures, must be a member of an `alternate_group` with a non-zero group identifier.

An alternate group is not needed if there is only one primary track, with a switching track. This switching track may contain SI pictures, or SP pictures for trick modes or error resilience, which predict both from and to the same track.

© ISO/IEC 2002 — All rights reserved

5.12.2 Track references

The switching track must be linked to the track into which it switches (the destination track) by a track reference of type 'swto' in the switching picture track.

If the switching track contains SP pictures, the switching track must be linked to the track from which it switches (the source track) by a track reference of type 'swfr' in the switching picture track.

5.12.3 Sample dependency

This subclause defines the dependencies of each switching sample on sample(s) in the source track. This table is only needed in a switching track which has a source ('swfr') track dependency.

5.12.3.1 Definition

Box Type: 'sdep'
Container: Sample Table 'stbl'
Mandatory: No
Quantity: Zero or exactly one.

This box contains the sample dependencies for each switching sample. The dependencies are stored in the table, one record for each sample. The size of the table, `sample_count` is taken from the `sample_count` in the Sample Size Box ('stsz').

5.12.3.1.1 Syntax

```
aligned(8) class SampleDependencyBox
   extends FullBox('sdep', version = 0, 0) {
   bit(1) multiple_representation_flag;
   bit(7) reserved = 0;
   for (i=0; i < sample_count; i++)
      if (multipleRep)
         representation_count = 1;
      else
         unsigned int(16)  representation_count;
      for (j=0; j < representation_count; j++) {
         unsigned int(16) dependency_count;
         for (k=0; k < dependency_count; k++) {
            signed int(16) relative_sample_number;
         }
      }
   }
}
```

5.12.3.1.2 Semantics

`mutiple_representation_flag` - is an flag that indicates whether each switching sample has multiple representations, each with its own separate dependency information.

`representation_count` - is an integer that a counts the number of representations in the switch sample. If

`dependency_count` - is an integer that a counts the number of samples in the source track on which this switching sample directly depends.

© ISO/IEC 2002 — All rights reserved

`relative_sample_number` - is an integer that identifies a sample in the source track. The relative sample numbers are encoded as follows. If there is a sample in the source track with the same decoding time, it has a relative sample number of 0. Whether or not this sample exists, the sample in the source track which immediately precedes the decoding time of the switching sample has relative sample number –1, the sample before that –2, and so on. Similarly, the sample in the source track which immediately follows the decoding time of the switching sample has relative sample number +1, the sample after that +2, and so on.

5.13 Representation of new structures in Movie Fragments

[TBD]

5.14 Use of the AVC File Format in MPEG-4 Systems

This subclause specifies how the AVC file format shall be used when the file is marked as being compatible with the MPEG-4 file format specified in ISO/IEC 14496-1:2001 Amendment 6. This subclause applies when the file is branded with the MPEG-4 file format brand of 'mp41' or 'mp42', and the AVC video data must be used in an MPEG-4 systems context.

5.14.1 Forming the Elementary Stream Descriptor

As is normal for MPEG-4 streams, the TrackID is related to the `ElementaryStreamID`, and the `SLConfigDescriptor` is generated following the rules for any MPEG-4 stream The format of the ES Descriptor is specified in chapter 13 of the MPEG-4 systems specification [ISO/IEC 14966:2001].

If the Elementary Stream Descriptor should contain any other descriptors than `SLConfigDescriptor` or `DecoderConfigDescriptor`, they are stored in the Sample Description as defined in subclause 5.9.1.1.

5.14.2 Forming the DecoderConfigDescriptor

The object type and stream type indications have the values defined in subclause 4.2 for AVC streams.

The `buffersizeDB`, `maxBitrate`, and `avgBitrate` fields can be filled by inspection of the VUI Sequence Parameters, if present in the sequence parameter set for the AVC stream.

The `DecoderSpecificInfo` is formed using the contents of the `AVCConfigurationBox`.

5.14.3 Profile and Level Indication

The MPEG-4 `ProfileLevelIndication` can be deduced from the AVC Profile and Level Indication found in the `AVCDecoderConfiguration`.

[Editor's note: Until the code-points are assigned, it's not clear whether this is a one-one mapping.]

5.14.4 Elementary Stream Descriptors

The use of multiple non-dependent elementary stream descriptors may *also* be indicated by the presence of more than one independent AVC video elementary stream descriptors in an object descriptor.

© ISO/IEC 2002 — All rights reserved

5.14.5 Switching Picture Tracks

Switching picture tracks are *not* MPEG-4 elementary streams and shall not be included within an MPEG-4 object descriptor.

5.15 Possible issues

5.15.1 Typical patterns of NAL Unit sizes etc.

This design could probably benefit from more knowledge of typical patterns of use and sizes etc. For example, repeating patterns etc. can be leveraged to reduce overhead, and the sizes of fields adjusted (e.g. the relative sample number). See, for example, N8708.

5.15.2 Do we need better visibility into where SEIs occur?

If an application wishes to answer "where are the SEI messages in this stream?" it must search all samples for those beginning with SEIs. We are unaware of a need for such a search, but if such need arises, some kind of indexing for the SEIs may be needed.

5.15.3 Do we need to allow parameter set updates in the OD stream?

The file format effectively allowing update of `DecoderSpecificInfo`, which might change parameters such as the visual width and height (though the file format normalizes these). Does MPEG-4 systems need to allow such re-configuration of existing streams, or should we express constraints on the changes in configuration permitted in an MPEG-4 context?

5.15.4 Will chunking be affected by Parameter Set changes?

All the samples in a chunk share a single configuration. If parameter sets change frequently, and this causes the configuration to change frequently, this would have a significant adverse impact on the size of the file-format sample tables. Typical patterns of use would again be of assistance here.

5.15.5 Layering and sub-sequences

This whole area is provisional before we get clarification on a number of issues, including, but not limited to:

i. Clarifying whether there is one division into layers and sub-sequences, or multiple;

ii. What 'layers' mean if sub-sequences may be selected from any layer, provided that their dependents have been decoded;

iii. Given that sub-sequences begin and end and then begin again, what the exact meaning of a 'dependency' is (on which instances?);

iv. What the use of sub-sequence identifiers is; whether they refer to definitions or instances, and if the latter, how the identifier space is managed in an indefinite duration sequence;

v. How the beginning and end of a sub-sequence can be identified;

© ISO/IEC 2002 — All rights reserved

vi. Whether dependencies are only direct, or transitive;

vii. What the meaning of the bit-rate etc. are, in the definition of sub-sequences, with respect to the individual instances of the sub-sequences.

© ISO/IEC 2002 — All rights reserved

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ❑ BLACK BORDERS
- ❑ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ❑ FADED TEXT OR DRAWING
- ❑ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ❑ SKEWED/SLANTED IMAGES
- ❑ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ❑ GRAY SCALE DOCUMENTS
- ❑ LINES OR MARKS ON ORIGINAL DOCUMENT
- ❑ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ❑ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

©ISO/IEC 2002 — All rights reserved

INTERNATIONAL ORGANISATION FOR STANDARDISATION

ORGANISATION INTERNATIONALE DE NORMALISATION

ISO/IEC JTC1/SC29/WG11

CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11

MPEG02/N5303

09 January 2003

Deleted: MPEG02/N5096

Title:	Study Text of ISO/IEC 14496-15/FCD
Authors:	David Singer, Toby Walker (Editors)
Group:	MPEG-4
Subgroup:	Systems

Deleted: Source: . MPEG-4 Systems¶
Title: . Study Text of Amd7 of ISO/IEC 14496-1 PDAM7 ¶
Status: . Study of FCD¶

Deleted: Authors: . David ¶
Singer, Toby Walker (Editors)¶
Group: . MPEG-4 ¶
Subgroup: . Systems ¶

Document type: International Standard
Document subtype: Amendment
Document stage: (30) Committee
Document language: E

Field Code Changed
Field Code Changed
Field Code Changed

ISO/IEC 14496-15:2002/StudyCD

ISO/IEC JTC 1/SC 29/WG 11

Secretariat:

Information technology — Coding of audio-visual objects — Part 15: AVC File Format

Élément introductif — Élément central — Partie 1: Titre de la partie

Error! AutoText entry not defined.

Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

[Indicate the full address, telephone number, fax number, telex number, and electronic mail address, as appropriate, of the Copyright Manger of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

© ISO/IEC 2002 — All rights reserved

Contents

Page

INTERNATIONAL ORGANISATION FOR STANDARDISATION 1
Foreword vi
Introduction vii
1 SCOPE 1
2 NORMATIVE REFERENCES 1
3 TERMS AND DEFINITIONS 1
3.1 Terminology 1
3.1.1 Parameter Set 1
3.1.2 Parameter Set Elementary Stream 1
3.1.3 Video Elementary Stream 1
3.2 Symbols and abbreviated terms 2
4 USE OF AVC IN MPEG-4 SYSTEMS 2
4.1 AVC Streams 2
4.1.1 Structure of the AVC streams 2
4.1.2 Time Base for AVC Streams 3
4.1.3 Access unit definition 4
4.1.4 AVC Access Unit Syntax and Semantics 5
4.1.5 Decoder Specific Information 6
4.1.6 AVC DecoderConfigDescriptor 8
4.1.7 Elementary Stream Descriptor 9
4.2 Codepoints 9
4.2.1 visualProfileLevelIndication 9
4.2.2 objectTypeIndication 10
4.3 Extensibility 11
5 File Format 11
5.1 Introduction 11
5.2 Terminology 11
5.3 AVC File type and identification 12
5.4 AVC Track Structure 12
5.5 Canonical order and restrictions 12
5.6 Sample format 13
5.7 Template Fields Used 13
5.8 Visual Width and Height 14
5.9 Sample Table elements 14
5.9.1 Sample description name and format 14
5.9.2 Decoding Time (DTS) and Composition Time (DTS) 15
5.9.3 Sync sample (IDR) 15
5.9.4 Shadow sync 15
5.10 Sample Groups 16
5.10.1 Introduction 16
5.11 Layering and sub-sequences 18
5.12 Alternate Streams and Switching Pictures 20
5.12.1 Alternate Group 21
5.12.2 Track references 22
5.12.3 Sample dependency 22

Field Code Changed

© ISO/IEC 2002 — All rights reserved

5.13 Representation of new structures in Movie Fragments 23
5.14 Use of the AVC File Format in MPEG-4 Systems 23
5.14.1 Forming the Elementary Stream Descriptor 23
5.14.2 Forming the DecoderConfigDescriptor 23
5.14.3 Profile and Level Indication 23
5.14.4 Elementary Stream Descriptors 23
5.14.5 Switching Picture Tracks 24
5.15 Possible issues 24
5.15.1 Typical patterns of NAL Unit sizes etc. 24
5.15.2 Do we need better visibility into where SEIs occur? 24
5.15.3 Do we need to allow parameter set updates in the OD stream? 24
5.15.4 Will chunking be affected by Parameter Set changes? 24
5.15.5 Layering and sub-sequences 24

© ISO/IEC 2002 — All rights reserved

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 14496-15 was prepared by ISO/IEC/TC JTC 1/SC 29.

Deleted: Amendment 7 to ISO/IEC 14496-1:2001

© ISO/IEC 2002 — All rights reserved

Introduction

The Advanced Video Codec (AVC), jointly developed by the ITU and ISO/IEC SC29/WG11 (MPEG), offers not only increased coding efficiency and enhanced robustness, but also interesting opportunities and challenges for the systems which use it. To enable the best visibility of, and access to, those features, and to enhance the opportunities for the interchange and interoperability of media, this standard defines a storage format for video streams compressed using AVC and using carrying within the MPEG-4 Systems Framework (ISO/IEC 14496-1:2001 and its various amendments).

This specification defines a storage format based on, and compatible with, the ISO Media File format ISO/IEC 14496-12 | 15444-12), which is used by the MP4 file format (ISO/IEC 14496-14) and the Motion JPEG 2000 file format (ISO/IEC 15444-1) among others. This specification enables AVC video streams to:

Deleted: (ISO/IEC 14496-1:2001, Amd. 5),

Deleted: 14496-1 Amd. 6)

- be used in conjunction with other media streams, such as audio;
- be used in an MPEG-4 systems environment, if desired;
- be formatted for delivery by a streaming server, using hint tracks;
- inherit all the use cases and features of the ISO Media file format on which MP4 and MJ2 are based.

The ISO Media File format is becoming increasingly common as a general-purpose media container format for the exchange of digital media, and its use in this context should accelerate both adoption and interoperability.

Extensions to the base format are defined here to expose the new systems aspects of the AVC codec.

© ISO/IEC 2002 — All rights reserved

Information technology — Coding of audio-visual objects — Part 15: AVC File Format

1 SCOPE

This document specifies the storage format for AVC (ISO/IEC 14496-10 | ITU Recommendation H.264) video streams. It also specifies how such streams are carried in an MPEG-4 context.

[Ed's Note: It is expected that the specification of carriage of AVC streams in MPEG-4 Systems will move into an appropriate MPEG-4 Systems Amendment].

2 NORMATIVE REFERENCES

[1] ISO/IEC 14496-1:2001, *Information Technology – Coding of audio-visual objects – Part 1 :Systems*
[2] ISO/IEC 14496-12, *Information Technology – Coding of audio-visual objects – Part 12 :ISO Media File Format*
[3] ISO/IEC 14496-14, *Information Technology – Coding of audio-visual objects – Part 14: MP4 File Format*
[4] ISO/IEC 14496-10, *Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding, Final Committee Draft, ISO/SC29/WG11/N4920.*

Formatted: Bullets and Numbering

3 TERMS AND DEFINITIONS

For the purposes of this specification, the terms and definitions specified in normative references ISO/IEC 14496-1 and ISO/IEC 14496-10 | ITU Recommendation H.264 shall apply. In addition, the following terms and abbreviations shall apply to all clauses in this specification.

3.1 Terminology

3.1.1 Parameter Set

Formatted: Bullets and Numbering

A parameter set is either a sequence parameter set or a picture parameter set. This term is used to refer to both types of parameter sets.

3.1.2 Parameter Set Elementary Stream

An elementary stream containing access units made up only of sequence and picture parameter set NAL units, synchronized with the video elementary stream.

Deleted: that is

3.1.3 Video Elementary Stream

An elementary stream containing access units made up of NAL units for coded picture data.

Deleted: containing video coding layer data, SEI messages, and (optionally) parameter set update NAL units.

© ISO/IEC 2002 — All rights reserved

3.2 Symbols and abbreviated terms

AVC	ISO/IEC 14496-10, Advanced Video Coding
NAL	Network Abstraction Layer
SEI	Supplementary Enhancement Information

4 USE OF AVC IN MPEG-4 SYSTEMS

4.1 AVC Streams

This subclause specifies the elementary stream format used to carry AVC visual content within the MPEG-4 systems framework.

4.1.1 Structure of the AVC streams

This subclause defines two elementary streams for carriage of AVC content within the MPEG-4 systems framework: a video elementary stream for carrying video data and an optional parameter set elementary stream for carrying parameter sets out-of-band from the video data.

AVC specifies a set of Network Abstraction Layer units, which contain different types of data. While the syntax of each NAL unit is specified by AVC, how NAL units are to be carried as elementary streams in the MPEG-4 Systems framework is not specified by ISO/IEC 14496-10. This subclause specifies the format of MPEG-4 elementary streams for AVC content for use within MPEG-4 Systems.

AVC content shall be carried in MPEG-4 Systems in either one or two elementary streams:

- **Video Elementary Stream**: This elementary stream, structured into video access units, contains video coding related NAL units (i.e. those NAL units containing video data or signaling video structure). If the optional parameter set elementary stream is not present, sequence and picture parameter set NAL units may be present in this stream.

Deleted:), SEI NAL units, and filler data NAL units

- **Parameter Set Elementary Stream**: This optional stream is used to transmit only sequence and picture parameter set NAL units. If this stream is used, sequence and picture parameter set NAL units shall not be carried in the video elementary stream.

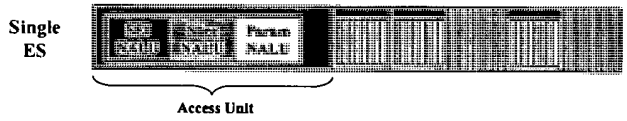
Formatted: Bullets and Numbering

The set types of NAL units are allowed in each of video and parameter set elementary streams is specified in Table 1.

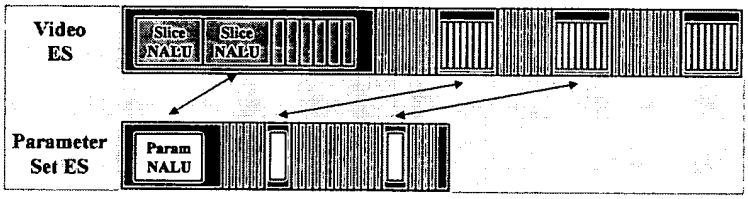
© ISO/IEC 2002 — All rights reserved

Table 1 -- NAL Unit Types in Elementary Streams

Value of nal_unit_type	Description	Video_ES	Parameter Set_ES
0x0	Reserved for external use	No	No
0x1	Coded slice	Yes	No
0x2	Coded data partition A (DPA)	Yes	No
0x3	Coded data partition B (DPB)	Yes	No
0x4	Coded data partition C (DPC)	Yes	No
0x5	Coded slice of an IDR picture	Yes	No
0x6	Supplemental Enhancement Information (SEI)	Yes	No
0x7	Sequence Parameter Set (SPS)	Yes, if parameter ES is not used.	Yes
0x8	Picture Parameter Set (PPS)	Yes, if parameter ES is not used.	Yes
0x9	Picture Delimiter (PD)	Yes	No
0xA	Filler Data (FD)	Yes	Yes
0xB – 0x17	Reserved	No	No
0x18 – 0x1F	For external use	Not specified by this standard.	Not specified by this standard.



(a) Single Video Elementary Stream containing all NAL Units



(b) Synchronized Video and Parameter with arrows denoting synchronization between streams.

Figure 1: AVC Elementary Stream Structure

4.1.2 Time Base for AVC Streams

The time base associated to an AVC stream shall be indicated using the means provided by MPEG-4 systems. If present, an AVC parameter set elementary stream shall adhere to the same time base as the AVC video elementary stream.

© ISO/IEC 2002 — All rights reserved

4.1.3 Access unit definition

An AVC access unit consists of one or more NAL units, as defined in ISO/IEC 14496-10. All NAL units that are to be processed at the same instant in time shall constitute a single access unit. Access units in AVC streams shall be labelled and time-stamped by suitable means. This shall be done via the related flags and the composition time stamps, respectively, in the SL packet header (see ISO/IEC 14496-1:2001, subclause 10.2.4). The composition time indicates the point in time at which the decoded data shall be composed. The decoding and composition time for an AVC access unit can have different values.

An AVC access unit shall set the `randomAccessPointFlag` in the SL packet header to '1' for this access if ALL of the following conditions are met:

- All video data NAL unit indicate a coded slice of an instantaneous decoder refresh (IDR) picture.
- All sequence and picture parameter sets needed to decode the video data in the access unit are contained in the decoder specific information for the video elementary stream.

Otherwise, the `randomAccessPointFlag` shall be set to '0'.

An AVC access unit shall include the following NAL units:

- **Video Data NAL Units (Coded Slice, Coded DPA, Coded DPB, Coded DPC , Coded IDR)**: All slice and data partition NAL units for a single picture shall be contained with the access unit whose decoding time and composition time is that of the picture. A single picture is one of a frame, a top and bottom field for one frame, or a single top or bottom field. Each AVC access unit shall contain at least one Video Data NAL unit.
- **SEI Message NAL Units**: All SEI NAL units shall be contained in the access unit whose decoding time is that before which the SEI messages come into effect instantaneously. This means that the SEI messages for a picture shall be included in the access unit containing to that picture and that SEI messages pertaining to a sequence of pictures shall be included in the access unit containing the first picture of the sequence to which the SEI message pertains. SEI messages that pertain to the picture level or higher levels shall occur prior to any video data NAL units in the access unit. Slice level messages may occur anywhere within an access unit.
- **Filler Data NAL Units:** Filler data NAL units can be included in any access unit. An access unit shall not consist entirely of filler data NAL units.
- **Picture Delimiter NAL Units**: Picture delimiter NAL units, if present, shall occur once at the beginning of the access unit whose decoding time is that of the picture being delimited.
- **Parameter Sets**: If a single video elementary stream is used to carry AVC content, all sequence and picture parameter sets NAL units shall be contained in the access unit whose decoding time is the time equal or prior to when the parameter sets come into effect. As specified in ISO/IEC 14496-10, any sequence or picture parameter sets present in an access unit shall occur prior to any video coding NAL units and SEI NAL units in that access unit. If a parameter set elementary steam is used, then the access unit in the parameter stream shall have a decoding time equal or prior to when the parameter set(s) comes into effect instantaneously. This means that for a parameter set to be used in a picture it must be sent prior to the access unit containing that picture or in the access unit for that picture.

The following restrictions apply to the Access Units:

© ISO/IEC 2002 — All rights reserved

a) **Timing SEIs**: The Temporal Reference SEI message and the clock timestamp SEI messages that define the timing information shall not be present in the video elementary stream as timing is provided by the MPEG-4 Sync Layer.

b) **Start Codes** shall not be present in the stream.

The sequence of NAL units in an access unit shall be any legal decoding order conforming to that specified in ISO/IEC 14496-10.

4.1.4 AVC Access Unit Syntax and Semantics

This subclause specifies the syntax and semantics of access units for AVC video. Access Units are externally framed and have a size supplied by that external framing. The structure of an AVC access unit is depicted in Figure 2.

Figure 2: The structure of an AVC Access Unit

An AVC access unit contains a succession of coded picture representations, each of which is made up of a set of NAL units. Each NAL units is is represented with a:

Deleted: NAL Units,

- *Length*: Indicates the length in bytes of the following NAL Unit. The length field can be configured to be of 1, 2, 3 or 4 bytes.
- *NAL Unit*: Contains the NAL unit data as specified in ISO/IEC 14496-10.

If a length field contains the value 0, then all remaining bytes in the access unit contain extension data which is reserved for use in future versions of this specification and must be ignored; those bytes are not NAL Units and must not be passed to the AVC decoder. No coded pictures or their data may occur after such a zero length field. In other words, extension data must be at the very end of the access unit.

Deleted: decoder.¶

4.1.4.1 Syntax

```
class aligned(8) AVCAccessUnit
{
  if(AVCConfigurationRecord.PictureLengthSize > 0)
    unsigned int(8)   PictureCount
  else
    PictureCount = 1;
  for(i=0; i<PictureCount; i++)
  {
    if(AVCConfigurationRecord.PictureLengthSize > 0)
      unsigned int(AVCConfigurationRecord.PictureLengthSize*8)
        PictureLength;
    for (i=0; ; i++;)    // to end of the picture
    {
      unsigned int((AVCConfigurationRecord.LengthSizeMinusOne+1)*8) NALUnitLength;
      if (NALUnitLength==0) {
```

Deleted: Access Unit .

© ISO/IEC 2002 — All rights reserved

```
        bit() reserved; // to end of the picture
        break;
      } else {
        bit(NALUnitLength * 8) NALUnit;
      }
    }
  }
}
```

Deleted: Access Unit .

4.1.4.2 Semantics

An AVC Video Access Unit consists of a set of NAL units with the same decoding and composition time. A single access unit can contain more than one picture if it contains multiple coded representations of the same picture (either redundant pictures or multiple switching pictures). The syntax of an access unit is configured via the decoder specific configuration for the AVC elementary stream. An AVC Parameter Set Access Unit contains one or more sequence and picture parameter set NAL units. If the coded representations of a picture are redundant pictures then the primary picture (redundant_pic_cnt = 0) shall occur first and all subsequence redundant picture shall occur in ascending order of their redundant picture (redundant_pic_cnt) value.

Deleted: update

Deleted: ¶

A video elementary stream is a random access point if and only if it contains an IDR picture and either of the following conditions holds:

Formatted: Bullets and Numbering

- All of the sequence and picture parameter sets necessary to decode the IDR picture; or,
- A separate parameter set elementary stream is used.

A parameter set elementary stream access unit is a random access point if and only if it contains all sequence and picture parameter sets used from that time forward in the video coding stream and parameter set stream until the next time any parameter set is updated.

`PictureCount` indicates the number of coded representations of the same picture contained within this unit. Such pictures may not be identical objective representations of the same picture. This field shall have a value of at least one.

`PictureLength` indicates the number of bytes contained the coded data for this picture's representation, which is made up of the set of NAL units preceded by length fields. The length of a picture comprises the number of bytes in all NAL units and the size of all `NALUnitLength` fields. If an access unit includes extension data then the picture length field does not include the size of the data or the `NALUnitLength` field whose value is zero.

`NALUnitLength` indicates the size of a NAL unit measured in bytes. The length field includes the size of both the one byte NAL header and the EBSP payload but does not include the length field itself.

`NALUnit` contains a single NAL unit. The syntax of an NAL unit is defined in ISO/IEC 14496-10 and includes both the one byte NAL header and the variable length encapsulated byte stream payload.

4.1.5 Decoder Specific Information

This subclause specifies the decoder specification information for ISO/IEC 14496-10 video content.

4.1.5.1 AVC Decoder Configuration Record

This record contains the size of the length field used in each access unit to indicate the length of its contained NAL units as well as the initial parameter sets. This record is externally framed (its size must be supplied by the structure which contains it).

© ISO/IEC 2002 — All rights reserved

This record contains a version field. This version of the specification defines version 1 of this record. Incompatible changes to the record will be indicated by a change of version number. Readers must not attempt to decode this record or the streams to which it applies if the version number is unrecognized.

Compatible extensions to this record will extend it and will not change the configuration version code. Readers should be prepared to ignore unrecognized data beyond the definition of the data they understand (e.g. after the parameter sets in this specification).

When used to provide the configuration of a parameter set elementary stream, the configuration record shall contain no sequence or picture parameter sets (`numOfParameterSets` shall have the value 0).

4.1.5.1.1 Syntax

```
aligned(8) class AVCConfigurationRecord {
   unsigned int(8) configurationVersion = 1;
   unsigned int(8) AVCProfileIndication;
   unsigned int(8) AVCLevelIndication;
   unsigned int(8) PictureLengthSize;
   unsigned int(16) nonStoredDegredationPriorityLow;
   unsigned int(16) nonStoredDegredationPriorityHigh;
   bit(6) reserved = '111111'b;
   unsigned int(2) lengthSizeMinusOne;
   unsigned int(8) numOfSequenceParameterSets;
   for (i=0; i< numOfSequenceParameterSets;  i++) {
      unsigned int(16) sequenceParameterSetLength ;
      bit(8*sequenceParameterSetLength) sequenceParameterSetNALUnit;
   }
   unsigned int(8) numOfPictureParameterSets;
   for (i=0; i< numOfPictureParameterSets;  i++) {
      unsigned int(16) pictureParameterSetLength;
      bit(8*pictureParameterSetLength) pictureParameterSetNALUnit;
   }
}
```

Deleted: parameterSetLength; .

Deleted: bit(8*parameterSetLength) parameterSetUpdateNALUnit; .

4.1.5.1.2 Semantics

`AVCProfileLevelIndication` contains the profile code defined in Annex A of the AVC specification [ISO/IEC 14496-10 | ITU Recommendation H.264].

`AVCLevelIndication` contains the level code defined in Annex A of the AVC specification [ISO/IEC 14496-10 | ITU Recommendation H.264].

`PictureLengthSize` indicates the length of the picture size field used to indicate the size of a coded picture representation. A value of zero indicates that only a single coded picture representation is allowed in an access unit.

`nonStoredDegrediationPriorityHigh` indicates the largest value of the range of degradation priorities that correspond to AVC non-stored frames (nal_type_idc=0). If the value of `nonStoredDegrediationPriorityHigh` exceeds `nonStoredDegrediationPriorityLow` then no correspondence between degradation priorities and non-stored frames is defined.

`nonStoredDegrediationPriorityLow` indicates the smallest value of the range of degradation priorities that correspond to AVC non-stored frames (nal_type_idc=0).

`lengthSizeMinusOne` indicates the length in bytes of the `NALUnitLength` field in an AVC video access unit of the associated stream minus one. For example, a size of one byte is indicated with a value of 0.

`numOfSequenceParameterSets` indicates the number of sequence parameter sets that are used as the initial set of sequence parameter sets for decoding the AVC elementary stream.

© ISO/IEC 2002 — All rights reserved

sequenceParameterSetLength indicates the length of the parameter set NAL unit as defined in ISO/IEC 14496-10.

sequenceParameterSetNALUnit contains a sequence parameter set NAL Unit, as specified in ISO/IEC 14496-10. Sequence parameter sets shall occur in order of ascending parameter set identifier with gaps being allowed.

numOfPictureParameterSets indicates the number of picture parameter sets that are used as the initial set of picture parameter sets for decoding the AVC elementary stream.

pictureParameterSetLength indicates the length of the parameter set NAL unit as defined in ISO/IEC 14496-10.

pictureParameterSetNALUnit contains a parameter set NAL Unit, as specified in ISO/IEC 14496-10. Picture parameter sets shall occur in order of ascending parameter set identifier with gaps being allowed.

4.1.5.2 AVC Stream DecoderSpecificInfo

This subclause defines the DecoderSpecificInfodescriptor for an AVC elementary stream.

4.1.5.2.1 Syntax

```
aligned(8) class AVCDecoderSpecificInfo extends DecoderSpecificInfo : bit(8)
tag=DecSpecificInfoTag {
    AVCVideoConfigurationRecord config;
}
```

4.1.5.2.2 Semantics

The decoder specific information for an AVC stream contains an AVC video stream decoder configuration record.

config contains the decoder configuration record for the AVC elementary stream decoder configuration.

4.1.5.3 Parameter Set

Initial sequence and picture parameter sets are supplied in the DecoderSpecificInfo for the AVC video stream, as defined in subclause 4.1.5.1. If a parameter set must be updated, then one of two methods can be used:

a) Sequence or Picture Parameter Set NAL units may be inserted in the video stream;

b) A separate parameter set stream may be used to carry parameter sets, possibly using a transport with different error resilience or reliability characteristics.

4.1.6 AVC DecoderConfigDescriptor

The DecoderConfigDescriptor shall set the value of streamtype equal to 0x04 for both AVC video elementary stream and the AVC parameter set elementary streams.

The DecoderConfigDescriptorfor an AVC video elementary stream, possibly including in-line sequence or picture parameter sets, shall set objectTypeIndicationto be 0x21 (ISO/IEC 14496-10 Visual). For the AVC parameter set elementary stream, the objectTypeIndication value shall equal 0x22 (ISO/IEC 14496-10 Visual Parameter Sets).

Deleted: parameterSetLength

Deleted: parameterSetUpdateNALUnit contains a parameter set update

Deleted: ¶
<#>AVC Stream DecoderSpecificInfo¶
This subclause defines the DecoderSpecificInfo descriptor for an AVC elementary stream.¶
<#>Syntax¶
aligned(8) class AVCDecoderSpecificInfo extends DecoderSpecificInfo : bit(8) tag=DecSpecificInfoTag { . . AVCVideoConfigurationRecord config; . } .
¶
<#>Semantics¶
The decoder specific information for an AVC stream contains an AVC video stream decoder configuration record.¶
config contains the decoder configuration record for the AVC elementary stream decoder configuration.¶
<#>Parameter Set Update¶
Initialparameter sets are supplied in the DecoderSpecificInfo for the AVC video stream, as defined in subclause 4.1.5.1. Ifparameter sets must be updated, then one of two methods can be used:¶
<#>Parameter Set NAL units may be inserted in the video stream;¶
<#>a separate parameter set stream may be used to carry parameter set updates, possibly using a transport with different error resilience or reliability characteristics. ¶
<#>AVC DecoderConfigDescriptor¶
The DecoderConfigDescriptor shall set the value of streamtype equal to 0x04 for both AVC video elementary stream and the AVC parameter set elementary streams. ¶
The DecoderConfigDescriptor for an AVC video elementary stream, possibly including in-lineparameter set updates, shall set objectTypeIndication to be 0x21 (ISO/IEC 14496-10 Visual). For the AVC parameter set elementary stream, the objectTypeIndication value shall equal 0x22 (ISO/IEC 14496-10 Visual Parameter Sets). ¶
<#>Elementary Stream Descriptor ¶
If the parameter set elementary stream is present, the elementary stream descriptors for the two streams shall satisfy the following conditions:¶
<#>The video elementary stream is dependent on the parameter set elementary stream and the ES_Descriptor for the video elementary stream shall have a streamDependenc ... [1]

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

© ISO/IEC 2002 — All rights reserved

Formatted: Bullets and Numbering

4.1.7 Elementary Stream Descriptor

If the parameter set elementary stream is present, the elementary stream descriptors for the two streams shall satisfy the following conditions:

Formatted: Bullets and Numbering

(1) The video elementary stream is dependent on the parameter set elementary stream and the `ES_Descriptor` for the video elementary stream shall have a `streamDependenceFlag` equal to true and indicate the ESID of the parameter in the `dependsOnESID` field. The `streamDependenceFlag` in the `ES_Descriptor` for the parameter set elementary stream shall be false.

(2) The elementary stream clocks for the parameter set elementary stream and the video elementary stream shall be the same and synchronized. The `OCRstreamflag` and `OCR_ES_Id` fields in the `ESDescriptor` for the video elementary stream and parameter set elementary streams shall be used to indicate that both streams share the same OCR.

4.2 Codepoints

4.2.1 visualProfileLevelIndication

Ed's Note: What should the visual profile level indication in the initial object descriptor be? Is it even necessary to add a new one? The list below does not seem up to date anyway and I do not recall any amendments to MPEG-4 Systems that touch this table.

Replace Table 6 in ISO/IEC 14496 with the following table:

Table 2 - visualProfileLevelIndication Values

Value	Profile	Level
0x00	Reserved for ISO use	-
0x01	Simple	L3
0x02	Simple	L2
0x03	Simple	L1
0x04	Simple Scalable	L2
0x05	Simple Scalable	L1
0x06	Core	L2
0x07	Core	L1
0x08	Main	L4
0x09	Main	L3
0x0A	Main	L2
0x0B	N-Bit	L2
0x0C	Hybrid	L2
0x0D	Hybrid	L1
0x0E	Basic Animated Texture	L2
0x0F	Basic Animated Texture	L1
0x10	Scalable Texture	L3
0x11	Scalable Texture	L2
0x12	Scalable Texture	L1
0x13	Simple Face Animation	L2
0x14	Simple Face Animation	L1

© ISO/IEC 2002 — All rights reserved

0x15	Advanced Visual Coding	-
0x15-0x7F	reserved for ISO use	-
0x80-0xFD	user private	-
0xFE	no visual profile specified	-

Formatted: Bullets and Numbering

4.2.2 objectTypeIndication

Replace Table 8 in ISO/IEC 14496 with the following table:

Table 3 - objectTypeIndication Values

Value	`ObjectTypeIndication` Description
0x00	Forbidden
0x01	Systems ISO/IEC 14496-1 [a]
0x02	Systems ISO/IEC 14496-1 [b]
0x03-0x1F	reserved for ISO use
0x20	Visual ISO/IEC 14496-2 [c]
0x21	Visual ISO/IEC 14496-10 [e]
0x22	Visual ISO/IEC 14496-10 [e]
0x23-0x3F	reserved for ISO use
0x40	Audio ISO/IEC 14496-3 [d]
0x41-0x5F	reserved for ISO use
0x60	Visual ISO/IEC 13818-2 Simple Profile
0x61	Visual ISO/IEC 13818-2 Main Profile
0x62	Visual ISO/IEC 13818-2 SNR Profile
0x63	Visual ISO/IEC 13818-2 Spatial Profile
0x64	Visual ISO/IEC 13818-2 High Profile
0x65	Visual ISO/IEC 13818-2 422 Profile
0x66	Audio ISO/IEC 13818-7 Main Profile
0x67	Audio ISO/IEC 13818-7 LowComplexity Profile
0x68	Audio ISO/IEC 13818-7 Scaleable Sampling Rate Profile
0x69	Audio ISO/IEC 13818-3
0x6A	Visual ISO/IEC 11172-2
0x6B	Audio ISO/IEC 11172-3
0x6C	Visual ISO/IEC 10918-1
0x6D - 0xBF	reserved for ISO use
0xC0 - 0xFE	user private
0xFF	no object type specified

© ISO/IEC 2002 — All rights reserved

[a] This object type shall be used for all streamTypes defined in ISO/IEC 14496-1 except IPMP streams.

[b] Includes associated Amendment(s) and Corrigendum(a).

[c] Includes associated Amendment(s) and Corrigendum(a). The actual object types are defined in ISO/IEC 14496-2 and are conveyed in the DecoderSpecificInfo as specified in ISO/IEC 14496-2, Annex K.

[d] Includes associated Amendment(s) and Corrigendum(a). The actual object types are defined in ISO/IEC 14496-3 and are conveyed in the DecoderSpecificInfo as specified in ISO/IEC 144963 subpart 1 subclause 6.2.1.

[e] Includes associated Amendment(s) and Corrigendum(a). The actual object types are defined in ISO/IEC 14496-10 and are conveyed in the DecoderSpecificInfo as specified in this admendment subclause 4.1.5.

4.3 Extensibility

The configuration description format may be extended in future versions of this specification. Similarly, the definition of an AVC Access Unit allows for extension by future versions of this specification.

5 File Format

5.1 Introduction

This subclause defines how to store ISO/IEC 14496-10 data in the MP4 file format. The storage of AVC content uses the existing capabilities of the MP4 file format but also defines new features to support the following features of the AVC codec in the MP4 file format:

- **Switching Pictures**: To enable switching between different coded streams and substitution of pictures within the same stream.
- **Enhanced Group of Pictures**: Provides a structuring of the dependencies of a group of pictures to provide both random access and flexible stream structure (e.g. temporal scalability and layering).
- **Parameter Sets**: The values needed to decode a video coding data are grouped together into a single picture parameter set, which is referenced from each slice.

5.2 Terminology

Table 4 summarizes the correspondences between the sets of terminology used in AVC Video, MPEG-4 Systems (ISO/IEC 14494-1), and MP4 (ISO/IEC 14496-1

Table 4: Correspondence of Terms in AVC, MPEG-4 Systems, and MP4

AVC	MPEG-4 Systems	MP4
-	Presentation	Movie
Stream	Elementary Stream	Track
Picture	Access Unit	Sample

© ISO/IEC 2002 — All rights reserved

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

5.3 AVC File type and identification

Formatted: Bullets and Numbering

Conformance with this specification is indicated by the presence of the brand 'jvt1' in the compatible-brands list of the `FileTypeBox` as defined in the ISO Media File Format specification [ISO 14496-1:2001, Amd. 5].

Readers conformant to this specification should read the file if this brand occurs in the compatible-brands list. Other structures and/or track types, defined in other specifications, may be present, and these may be ignored by a conformant reader.

If the major-brand is 'jvt1', then

Formatted: Bullets and Numbering

a) the minor-version, as defined in this version of this specification, shall be 1.

b) the file extension shall be ".jvt" [to be checked for collision]

c) the mime type used to serve the file shall be "video/jvt" [to be allocated]

5.4 AVC Track Structure

In the terminology of the ISO Media File Format Specification, an AVC track is a video or visual track. It therefore uses:

Formatted: Bullets and Numbering

a) a `handler_type` of `'vide'` in the `HandlerBox`;

b) a video media header `'vmhd'`;

c) and, as defined below, a derivative of the `VisualSampleEntry`.

5.5 Canonical order and restrictions

The AVC elementary stream is stored in the MP4 file format in a *canonical* format. The canonical format is as *neutral* as possible so that systems that need to customize the stream for delivery over different transport protocols – MPEG-2 Systems, RTP, and so on — should not have to *remove* information from the stream while being free to *add* to the stream. Furthermore, a canonical format allows such operations to be performed against a known initial state.

The canonical stream format is an AVC video stream that satisfies the following conditions:

Formatted: Bullets and Numbering

- *No parameter sets in the video elementary stream.* All sequence and picture parameter set information shall be stored in the sample descriptions for the track or in a separate parameter set track containing a parameter set elementary stream. This ensures that it is not necessary to examine every part of the elementary stream to find relevant parameter sets. It also avoids dependencies of indefinite duration between the sample that contains the parameter set definition and the samples that use it. Parameter sets may be inserted into the video elementary stream when the file is streamed over a transport that permits such parameter set updates.

- *Parameter Set Track:* If present, a parameter set track shall be constructed so that all parameter set samples are random access points. This implies that each there shall be a parameter set access unit at each point a parameter set is updated. Each parameter set shall contain exactly the sequence and picture parameter sets needed to decode

Note: The use of a parameter track in the file format does not require that a system delivering AVC content a use a separate elementary stream for parameter sets. Instead, implementations may choose to map parameter sets to

© ISO/IEC 2002 — All rights reserved

in-band parameter set NAL units in the video elementary stream or use some out-of-band delivery mechanism defined by the transport layer.

- *All timing information is external to stream.* Timing SEI messages that define presentation or composition timestamps shall not be included in the AVC elementary stream. All timing information is stored in the various sample metadata tables. If timing information were embedded in the stream, then there would be a great danger of inconsistency, and a key feature of MP4, that editing can take place without modifying elementary streams, would break.

- *All SEI messages are at the beginning of an AU.* NAL units included in an access unit shall be ordered so that all SEI messages occur before all video NAL units (slices or data partitions) in the access unit. This allows a rapid inspection of the sample to determine whether it contains SEI messages. All SEI messages shall be contained in a single SEI message NAL unit.

- *Redundant Picture*: NAL units within a single access shall be ordered in non-decreasing order of redundant picture count (redundant_pic_cnt).

- *Slice Groups*: NAL units within a single picture (the same value of picture order count) shall be ordered in non-decreasing order of slice group identifier. Within the same slice group, slices shall be ordered by their first Macroblock location (first_mb in the slice header).

- *Partition Ordering:* All data partitions of a slice shall occur in the order: 'A' partition, 'B' partition, and 'C' partition. No interleaving of partitions from different slices is allowed.

- *No start codes.* The elementary stream shall not include start codes. As stored, each NAL Unit is preceded by a length field; this enables easy scanning of the sample's NAL Units. Systems that use start-codes will reformat the stream to use start codes.

- *Picture delimiter NAL units.* Picture delimiter NAL units may be present in the sample data. If present, they shall occur as the first NAL unit in a sample.

- *No filler data.* Video data is naturally represented as VBR in the file format and should be filled for transmission if needed. Filler NAL units and Filler SEI messages shall not be present in the file format stored stream.

Note that these restrictions are more restrictive than the use of AVC in MPEG-4 systems.

5.6 Sample format

A file format sample is a picture in AVC terms, and an access unit in MPEG-4 terms. The sample data shall follow the syntax and semantics defined for AVC video access units in subclause 4.1.4. Note, this format is used for storing data only and does not require the use of MPEG-4 systems framework to carry AVC visual data.

5.7 Template Fields Used

The ISO Media File Format defines a number of fields which have default values but which may be defined for use by specific sub-systems. Tracks containing AVC data use the following template fields:

a) `alternate_group` in the `TrackHeaderBox` (see subclause 5.12 on stream switching);

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

© ISO/IEC 2002 — All rights reserved

5.8 Visual Width and Height

The width and height fields in a `VisualSampleEntry` must correctly document the encoded width and height as indicated by the sequence parameter sets in the AVC stream that is described by that entry. These *must* correspond to the picture size established by the vui_pic_parameters, which includes optional frame cropping information. The width and height fields do *not* reflect any changes in size caused by SEI messages such as Pan-scan. The visual handling of SEI messages such as pan-scan is both optional and terminal-dependent. If the width and height of the sequence changes, then a new sample description is needed.

The width and height fields in the track header may not be the same as the width and height fields in the one or more `VisualSampleEntry` in the video track. As specified in the ISO Media File Format, if normalized visual presentation is needed, all the sequences are normalized to the track width and height for presentation.

Formatted: Bullets and Numbering

5.9 Sample Table elements

5.9.1 Sample description name and format

5.9.1.1 Sample Description Box

5.9.1.1.1 Definition

Box Types: 'jvt1', 'jvtC', 'm4ds'
Container: Sample Table Box ('stbl')
Mandatory: Yes
Quantity: Exactly one

An AVC visual sample entry shall contain an AVC Visual Stream Configuration Box, as defined below. This includes a `AVCVideoDecoderConfigurationRecord` as defined in subclause 4.1.5.1 as the content of `DecoderSpecificInfo` for MPEG-4.

Extension descriptors that should be inserted into the Elementary Stream Descriptor, when used in MPEG-4, may also be present.

Multiple sample descriptions may be used, as permitted by the ISO Media File format specification, to indicate sections of video that use different configurations of parameter sets.

Formatted: Bullets and Numbering

5.9.1.1.2 Syntax

```
   // Visual Sequences
class AVCConfigurationBox extends Box('jvtC') {
   AVCVideoConfigurationRecord() AVCConfig;
}

class MPEG4DescriptorsBox extends Box('m4ds') {
   Descriptor Descr[0 .. 255];
}

class AVCSampleEntry() extends VisualSampleEntry ('jvt1'){
   AVCConfigurationBox config;
   MPEG4ExtensionDescriptorsBox (); // optional
}
```


© ISO/IEC 2002 — All rights reserved

5.9.1.1.3 Semantics

Formatted: Bullets and Numbering

`Compressorname` in the base class `VisualSampleEntry` indicates the name of the compressor used with the value `"\016JVT/AVC Coding"` being recommended (\016 is 14, the length of the string as a byte)

`config` is defined in subclause 4.1.5.2.

`Descr` is a descriptor which should be placed in the `ElementaryStreamDescriptor` when this stream is used in an MPEG-4 context. This does not include `SLConfigDescriptor` or `DecoderConfigDescriptor`, but includes the other descriptors in order to be placed after the `SLConfigDescriptor`. The use of this field is specified in subclause 5.14.1.

Formatted: Bullets and Numbering

5.9.1.2 Parameter Sets

Each AVC sample description, which contains the AVC video stream decoder specific information, includes a group of sequence and picture parameter sets. This group of parameter sets functions much like a codebook. Each parameter set has an identifier, and each slice references the parameter set it was coded against using the parameter set's identifier.

In the file format each configuration of parameter sets is represented separately. A parameter set cannot be updated without causing a different sample description to be used. For example, suppose that samples 1-4 use picture parameter sets identified as 1, 2, 3 and a single sequence parameter set identified as 1. At sample 5 a new value of picture parameter set 2 is required but picture parameter sets 1 and 3 remain unaltered and are used until sample 10. In this case, the sample description for samples 1 through 4 is the same and contains the initial values of picture parameter sets 1, 2, 3 and sequence parameter set 1. At sample 5 the sample description must change to a second sample description, which contains the updated value for picture parameter set 2 *as well as* the original values of picture parameter sets 1 and 3 and sequence parameter set 1. This second sample description is used for samples 5 through 10.

Systems wishing to send sequence or picture parameter set updates will need to compare the two configurations to find the differences in order to send the appropriate parameter set updates.

Formatted: Bullets and Numbering

5.9.2 Decoding Time (DTS) and Composition Time (DTS)

Samples are stored in the file format in decoding order. If frame reordering is not used, this is also presentation order, and only the time-to-sample 'stts' table is used. Note that any kind of picture may be reordered in AVC video, not only B-frames.

If decoding time must be stored, the composition time-to-sample 'ctts' is also used.

Formatted: Bullets and Numbering

5.9.3 Sync sample (IDR)

A sync sample shall correspond to an instantaneously decodable picture in the AVC stream (also known as an Instantaneously Decodable Refresh point (IDR)).

Formatted: Bullets and Numbering

5.9.4 Shadow sync

The use of the shadow sync table to indicate alternate encodings of a sample for random access are supported for as defined in the ISO Base Media File format. A shadow sync shall indicate a sample that is an independently decodable sample, which shall be encoded as either an I-frame or a SI-frame.

While the use of shadow sync for AVC content is supported for backward compatibility reasons, this use is deprecated and use of the mechanisms defined in subclause 5.12 is recommended.

© ISO/IEC 2002 — All rights reserved

5.10 Sample Groups

Formatted: Bullets and Numbering

5.10.1 Introduction

This subclause specifies a generic mechanism for representing a partition of the samples in a track. A *sample grouping* is an assignment of each sample in a track to be a member of one *sample group*, based on a grouping criterion. A sample group in a sample grouping is not limited to being contiguous samples and may contain non-adjacent samples. As there may be more than one sample grouping for the samples in track, each sample grouping has a type field to indicate the type of grouping. For example, a file might contain two sample groupings for the same track: one based on an assignment of sample to layers and another to sub-sequences.

Sample groupings are represented by two linked data structures: (1) a `SampleToGroup` box represents the assignment of samples to sample groups; (2) a `SampleGroupDescription` box contains a *sample group entry* for each sample group describing the properties of the group. There may be multiple instances of the `SampleToGroup` and `SampleGroupDescription` boxes based on different grouping criteria. These are distinguished by a type field used to indicate the type of grouping.

One example of using these tables is to represent the assignments of samples to *layers*. In this case each sample group represents one layer, with an instance of the `SampleToGroup` box describing which layer a sample belongs to. For more details, please refer to subclause 5.11

Formatted: Bullets and Numbering

5.10.1.1 SampleToGroup Box

5.10.1.1.1 Definition

Box Type: 'sbgp'
Container: Sample Table Box ('stbl')
Mandatory: No
Quantity: Zero or more.

This table can be used to find the group that a sample belongs to and the associated description of that sample group. The table is compactly coded with each entry giving the index of the first sample of a run of samples with the same sample group descriptor. The sample group description ID is an index that refers to a `SampleGroupDescription` box, which contains entries describing the characteristics of each sample group.

There may be mutiple instance of this box if there is more the one sample grouping for the samples in a track. Each instance of the `SampleToGroup` box has a type code that distinguishes different sample groupings. Within a track, there shall be a most instance of this box with a particular grouping type. The associated `SampleGroupDescription` shall indicate the same value for the grouping type.

Formatted: Bullets and Numbering

5.10.1.1.2 Syntax

```
aligned(8) class SampleToGroupBox
   extends FullBox('sbst', version = 0, 0)
{
   unsigned int(32)  grouping_type;
   unsigned int(32)  entry_count;
   for (i=1; i <= entry_count; i++)
```


© ISO/IEC 2002 — All rights reserved

```
    {
        unsigned int(32)  sample_count;
        unsigned int(32)  group_description_index;
    }
}
```

Formatted: Bullets and Numbering

5.10.1.1.3 Semantics

`version` is an integer that specifies the version of this box.

`grouping_type` is an integer that identifies the type (i.e. criterion used to form the sample groups) of the sample grouping and links it to its sample group description table with the same value for grouping type. At most once occurrence of this box with the same value for `grouping_type` value shall exist for a track.

`entry_count` is an integer that gives the number of entries in the following table.

`sample_count` is an integer that gives the number of consecutive samples in the group.

`group_description_index` is an integer that gives the index of the sample group entry which describes the samples in this group. The index ranges from 1 to the number of sample group entries in the `SampleGroupDescription` Box.

Formatted: Bullets and Numbering

5.10.1.2 Sample Group Description Box

5.10.1.2.1 Definition

Box Types: 'sgpd'
Container: Sample Table Box ('stbl')
Mandatory: No
Quantity: Zero or more, with one for each Sample Group Table box.

This description table gives information about the characteristics of sample groups. The descriptive information is any other information needed to define or characterize the sample group.

There may be mutiple instance of this box if there is more the one sample grouping for the samples in a track. Each instance of the `SampleGroupDescription` box has a type code that distinguishes different sample groupings. Within a track, there shall be a most instance of this box with a particular grouping type. The associated `SampleToGroup` shall indicate the same value for the grouping type.

The information is stored in the sample group description box after the entry-count. An abstract entry type is defined and sample groupings shall define derived types to represent the description of each sample group. For video tracks, an abstract `VisualSequenceEntry` is used with similar types for audio and hint tracks.

Formatted: Bullets and Numbering

5.10.1.2.2 Syntax

```
// Sequence Entry
abstract class SampleGroupDesriptionEntry (unsigned int(32) handler_type)
{
}

// Visual Sequence
abstract class VisualSampleGroupEntry (type) extends SampleGroupDesriptionEntry
(type)
```

© ISO/IEC 2002 — All rights reserved

```
{
}

// Audio Sequences
abstract class AudioSampleGroupEntry (type) extends SampleGroupDesriptionEntry(type)
{
}

aligned(8) class SampleGroupDescriptionBox (unsigned int(32) handler_type)
   extends FullBox('sgpd', 0, 0){
   unsigned int(32) grouping_type;
   unsigned int(32) entry_count;
   int i;
   for (i = 1 ; i <= entry_count ; i++){
      switch (handler_type){
         case 'vide': // for video tracks
            VisualSampleGroupEntry ();
            break;
         case 'soun': // for audio tracks
            AudioSampleGroupEntry();
            break;
         case 'hint': // for hint tracks
            HintSampleGroupEntry();
            break;
      }
   }
}
```

Formatted: Bullets and Numbering

5.10.1.2.3 Semantics

`version` is an integer that specifies the version of this box.

`grouping_type` is an integer that identifies the `SampleToGroup` box that is associated with this sample group description.

`entry_count` is an integer that gives the number of entries in the following table.

Formatted: Bullets and Numbering

5.11 Layering and sub-sequences

Streams may be constructed so that the referential dependencies between samples allow only subsets of the samples to be sent to the decoder. This mechanism is called *thinning* a stream and is entirely transparent to the decoder receiving the elementary stream. Thinning discards entire sets of samples using knowledge of what other sets of pictures this set of pictures depends on and what picture sets in turn depend on it.

The referential dependencies between samples in a stream are structured into *layers* and *sub-sequences*. Samples in higher layers can only depend on samples in lower layers. Layers are numbered, and the samples organized such that a sample in layer N has no dependencies on samples in layers greater than N.

© ISO/IEC 2002 — All rights reserved

Sub-sequences are sets of samples that have external referential dependencies on other sets of samples. Dependency relations between sub-sequences represent the dependency structure of a stream. The samples in a sub-sequence can be disposed of as a unit when thinning streams. Each sub-sequence belongs to one and only one layer. A sample shall reside in one layer and in one sub-sequence only.

Layering and sub-sequences are not features of the codec but this information is represented in the file format to allow systems reading the files to understand the ways in which stream thinning may be achieved without having to examine the dependency structure of every sample.

Formatted: Bullets and Numbering

5.11.1.1 Sub-Sequence Description Entry

Box Types: 'jvss'
Container: Sample Group Description Box ('sgpd')
Mandatory: No
Quantity: Zero or more.

A sub-sequence description entry is a sample group entry that describes a sub-sequence. A sub-sequence is a set of samples in a track belonging to the same layer. A sub-sequence depends on another sub-sequence if and only if there exists a sample in the this sub-sequence that is directly referentially dependent on some sample in the other sub-sequence. All sample in a sub-sequence shall be depend only on other samples within the same sub-sequence or samples in the sub-sequences on which is it dependent. A sub-sequence can depend on zero or more sub-sequences in the same or lower layers. A sub-sequence shall not depend on sub-sequences in the higher scalability layers.

Formatted: Bullets and Numbering

5.11.1.1.1 Syntax

```
aligned(8) class DependencyInfo
{
    unsigned int(8)   layerNumber;
    unsigned int(32)  subSequenceIdentifier;
}

class AVCSubSequenceEntry () extends VisualSampleGroupEntry ('jvss')
{
    unsigned int(32) subSequenceIdentifer;
    unsigned int(32) avgBitRate;
    unsigned int(32) avgFrameRate;
    unsigned int(16) numReferences;
    DependencyInfo dependency[numReferences];
}
```

Formatted: Bullets and Numbering

5.11.1.1.2 Semantics

`layerNumber` and `subSequenceIdentifier` within the `DependencyInfo` class identify a sub-sequence that is used as a reference for this sub-sequence. Only direct, not indirect, referential dependencies shall be identified.

`subSequenceIdentifier` in the Sub-Sequence Box gives the identifier for the sub-sequence.

`avgBitRate` gives the average bit rate in bits/second of the sub-sequence within the segment. All NAL units belonging to pictures in the layer are taken into account in the calculation. A value of zero means an undefined bit rate.

`avgFrameRate` gives the average frame rate in frames/(256 seconds) of the sub-sequence within the segment. A value of zero means an undefined frame rate.

© ISO/IEC 2002 — All rights reserved

numReferences gives the number of sub-sequences directly referenced in this sub-sequence. dependency is an array of DependencyInfo structures giving the identification information of the referenced sub-sequences.

5.11.1.2 Layer Description Entry

Formatted: Bullets and Numbering

Box Types: 'jvll'
Container: Sample Group Description Box ('sgpd')
Mandatory: No
Quantity: Zero or more.

A layer sample group entry defines the layer information for all samples in a layer. Layers are ordered hierarchically based on their dependency on each other: The first or base layer is independently decodable. The second and subsequent layers are enhancement layers that depend on some data in the base layer. Pictures in a higher layer can only depend on pictures in lower layers.

5.11.1.2.1 Syntax

Formatted: Bullets and Numbering

```
class AVCLayerEntry() extends VisualSampleGroupEntry ('jvll')
{
  unsigned int(8)  layerNumber;
  unsigned int(32) avgBitRate;
  unsigned int(32) avgFrameRate;
}
```

5.11.1.2.2 Semantics

Formatted: Bullets and Numbering

layerNumber gives the number of this layer with the base layer being numbered as zero and all enhancement layers being numbered as one or higher with consecutive numbers.

avgBitRate gives the average bit rate in bits/second of the layer within the segment. A value of zero indicates an undefined bit rate.

avgFrameRate gives the average frame rate in frames per 1000 seconds of this layer with a value of zero indicating an undefined frame rate.

Formatted: Bullets and Numbering

5.12 Alternate Streams and Switching Pictures

In typical streaming scenarios, one of the key requirements is to scale the bit rate of the compressed data in response to changing network conditions. The simplest way to achieve this is to encode multiple streams wi th different bandwidths and quality settings for representative network conditions. The server can then switch amongst these pre-coded streams in response to network conditions. In current standards, switching between streams is only possible at I-frames, because the frames can only be switched when there are no dependencies on prior pictures for reconstruction.

AVC has adopted SP-pictures and SI-pictures ("switching pictures") that allow switching from one stream to another while still supporting inter-frame coding of switching pictures. Figure 3 shows how SP frames are used to switch between two different bit streams.

© ISO/IEC 2002 — All rights reserved

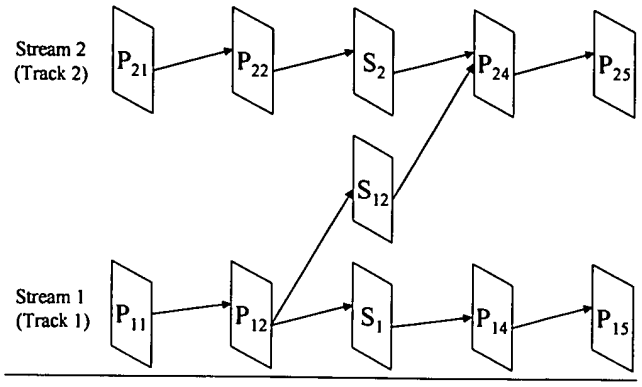


Figure 3: Stream Switching

In the file format, switching pictures are stored in *switching picture* tracks, which are tracks separate from the track that is being switched from and the track being switched to. Switching picture tracks can be identified by the existence of a specific required track reference in that track. A switching picture is an alternative to the sample in the destination track that has exactly the same decoding time. If all switching pictures are SI pictures, then no further information is needed.

If any of the pictures in the switching track are SP pictures, then two extra pieces of information may be needed. First, the source track, that is being switched from, must be identified by using a track reference (the source track may be the same track as the destination track). Second, the dependency of the switching picture on the samples in the source track may be needed, so that a switching picture is only used when the pictures on which it depends have been supplied to the decoder.

This dependency is represented by means of an optional extra sample table. There is one entry per sample in the switching track. Each entry records the relative sample numbers in the source track on which the switching picture depends. If this array is empty for a given sample, then that switching sample contains an SI picture. If the dependency box is not present, then only SI-frames shall be present in the track.

A switching sample may have multiple coded representations with different dependencies. For AVC video, the multiple representations are stored in the same sample (i.e. access unit) using the existing facility to have multiple coded representations in the same access unit. For example, one switch sample might contain a SP-picture representation dependent on some earlier samples, used for stream switching, and another representation as an SI-picture, used for random access.

Formatted: Bullets and Numbering

5.12.1 Alternate Group

The ISO media file format (but not the version one specification of the MPEG-4 file format, which is branded as 'mp41') supports the use of what is called *alternate tracks*. Each track can optionally specify an *alternate group* (in the track header box) that groups together alternate encodings of the same content. Thus, each alternate bit-stream can be stored as a separate track in the MP4 and related together as alternate tracks. All the tracks which form a group which may be switched between, but not the tracks containing the switching pictures, must be a member of an `alternate_group` with a non-zero group identifier.

An alternate group is not needed if there is only one primary track, with a switching track. This switching track may contain SI pictures, or SP pictures for trick modes or error resilience, which predict both from and to the same track.

© ISO/IEC 2002 — All rights reserved

Formatted: Bullets and Numbering

5.12.2 Track references

The switching track must be linked to the track into which it switches (the destination track) by a track reference of type 'swto' in the switching picture track.

If the switching track contains SP pictures, the switching track must be linked to the track from which it switches (the source track) by a track reference of type 'swfr' in the switching picture track.

Formatted: Bullets and Numbering

5.12.3 Sample dependency

This subclause defines the dependencies of each switching sample on sample(s) in the source track. This table is only needed in a switching track which has a source ('swfr') track dependency.

Formatted: Bullets and Numbering

5.12.3.1 Definition

Box Type: 'sdep'
Container: Sample Table 'stbl'
Mandatory: No
Quantity: Zero or exactly one.

This box contains the sample dependencies for each switching sample. The dependencies are stored in the table, one record for each sample. The size of the table, `sample_count` is taken from the `sample_count` in the Sample Size Box ('stsz').

Formatted: Bullets and Numbering

5.12.3.1.1 Syntax

```
aligned(8) class SampleDependencyBox
   extends FullBox('sdep', version = 0, 0) {
   bit(1) multiple_representation_flag;
   bit(7) reserved = 0;
   for (i=0; i < sample_count; i++)
      if (multipleRep)
         representation_count = 1;
      else
         unsigned int(16) representation_count;
      for (j=0; j < representation_count; j++) {
         unsigned int(16) dependency_count;
         for (k=0; k < dependency_count; k++) {
            signed int(16) relative_sample_number;
         }
      }
   }
}
```

Formatted: Bullets and Numbering

5.12.3.1.2 Semantics

`mutiple_representation_flag` - is an flag that indicates whether each switching sample has multiple representations, each with its own separate dependency information.

`representation_count` - is an integer that a counts the number of representations in the switch sample. If

`dependency_count` - is an integer that a counts the number of samples in the source track on which this switching sample directly depends.

© ISO/IEC 2002 — All rights reserved

`relative_sample_number` - is an integer that identifies a sample in the source track. The relative sample numbers are encoded as follows. If there is a sample in the source track with the same decoding time, it has a relative sample number of 0. Whether or not this sample exists, the sample in the source track which immediately precedes the decoding time of the switching sample has relative sample number –1, the sample before that –2, and so on. Similarly, the sample in the source track which immediately follows the decoding time of the switching sample has relative sample number +1, the sample after that +2, and so on.

5.13 Representation of new structures in Movie Fragments

[TBD]

5.14 Use of the AVC File Format in MPEG-4 Systems

This subclause specifies how the AVC file format shall be used when the file is marked as being compatible with the MPEG-4 file format specified in ISO/IEC 14496-1:2001 Amendment 6. This subclause applies when the file is branded with the MPEG-4 file format brand of 'mp41' or 'mp42', and the AVC video data must be used in an MPEG-4 systems context.

5.14.1 Forming the Elementary Stream Descriptor

As is normal for MPEG-4 streams, the TrackID is related to the `ElementaryStreamID` and the `SLConfigDescriptor`is generated following the rules for any MPEG-4 stream The format of the ES Descriptor is specified in chapter 13 of the MPEG-4 systems specification [ISO/IEC 14966:2001].

If the Elementary Stream Descriptor should contain any other descriptors than `SLConfigDescriptor` or `DecoderConfigDescriptor` they are stored in the Sample Description as defined in subclause 5.9.1.1.

5.14.2 Forming the DecoderConfigDescriptor

The object type and stream type indications have the values defined in subclause 4.2 for AVC streams.

The `buffersizeDB`, `maxBitrate`, and `avgBitrate` fields can be filled by inspection of the VUI Sequence Parameters, if present in the sequence parameter set for the AVC stream.

The `DecoderSpecificInfo`is formed using the contents of the `AVCConfigurationBox`

5.14.3 Profile and Level Indication

The MPEG-4 `ProfileLevelIndication`can be deduced from the AVC Profile and Level Indication found in the `AVCDecoderConfiguration`

[Editor's note: Until the code-points are assigned, it's not clear whether this is a one-one mapping.]

5.14.4 Elementary Stream Descriptors

The use of multiple non-dependent elementary stream descriptors may *also* be indicated by the presence of more than one independent AVC video elementary stream descriptors in an object descriptor.

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

© ISO/IEC 2002 — All rights reserved

5.14.5 Switching Picture Tracks

Formatted: Bullets and Numbering

Switching picture tracks are *not* MPEG-4 elementary streams and shall not be included within an MPEG-4 object descriptor.

5.15 Possible issues

Formatted: Bullets and Numbering

5.15.1 Typical patterns of NAL Unit sizes etc.

Formatted: Bullets and Numbering

This design could probably benefit from more knowledge of typical patterns of use and sizes etc. For example, repeating patterns etc. can be leveraged to reduce overhead, and the sizes of fields adjusted (e.g. the relative sample number). See, for example, N8708.

5.15.2 Do we need better visibility into where SEIs occur?

Formatted: Bullets and Numbering

If an application wishes to answer "where are the SEI messages in this stream?" it must search all samples for those beginning with SEIs. We are unaware of a need for such a search, but if such need arises, some kind of indexing for the SEIs may be needed.

5.15.3 Do we need to allow parameter set updates in the OD stream?

Formatted: Bullets and Numbering

The file format effectively allowing update of `DecoderSpecificInfo` which might change parameters such as the visual width and height (though the file format normalizes these). Does MPEG-4 systems need to allow such re-configuration of existing streams, or should we express constraints on the changes in configuration permitted in an MPEG-4 context?

5.15.4 Will chunking be affected by Parameter Set changes?

Formatted: Bullets and Numbering

All the samples in a chunk share a single configuration. If parameter sets change frequently, and this causes the configuration to change frequently, this would have a significant adverse impact on the size of the file-format sample tables. Typical patterns of use would again be of assistance here.

5.15.5 Layering and sub-sequences

Formatted: Bullets and Numbering

This whole area is provisional before we get clarification on a number of issues, including, but not limited to:

Formatted: Bullets and Numbering

i. Clarifying whether there is one division into layers and sub-sequences, or multiple;

ii. What 'layers' mean if sub-sequences may be selected from any layer, provided that their dependents have been decoded;

iii. Given that sub-sequences begin and end and then begin again, what the exact meaning of a 'dependency' is (on which instances?);

iv. What the use of sub-sequence identifiers is; whether they refer to definitions or instances, and if the latter, how the identifier space is managed in an indefinite duration sequence;

v. How the beginning and end of a sub-sequence can be identified;

© ISO/IEC 2002 — All rights reserved

vi. Whether dependencies are only direct, or transitive;

vii. What the meaning of the bit-rate etc. are, in the definition of sub-sequences, with respect to the individual instances of the sub-sequences.

© ISO/IEC 2002 — All rights reserved

4.1.5.2AVC Stream DecoderSpecificInfo

This subclause defines the `DecoderSpecificInfo` descriptor for an AVC elementary stream.

4.1.5.2.1Syntax

```
aligned(8) class AVCDecoderSpecificInfo extends DecoderSpecificInfo : bit(8)
tag=DecSpecificInfoTag {
        AVCVideoConfigurationRecord config;
}
```

4.1.5.2.2Semantics

The decoder specific information for an AVC stream contains an AVC video stream decoder configuration record.

`config` contains the decoder configuration record for the AVC elementary stream decoder configuration.

4.1.5.3Parameter Set Update

Initialparameter sets are supplied in the `DecoderSpecificInfo` for the AVC video stream, as defined in subclause 4.1.5.1. Ifparameter sets must be updated, then one of two methods can be used:

a)Parameter Set NAL units may be inserted in the video stream;

b)a separate parameter set stream may be used to carry parameter set updates, possibly using a transport with different error resilience or reliability characteristics.

4.1.6AVC DecoderConfigDescriptor

The `DecoderConfigDescriptor` shall set the value of `streamtype` equal to 0x04 for both AVC video elementary stream and the AVC parameter set elementary streams.

The `DecoderConfigDescriptor` for an AVC video elementary stream, possibly including in-lineparameter set updates, shall set `objectTypeIndication` to be 0x21 (ISO/IEC 14496-10 Visual). For the AVC parameter set elementary stream, the `objectTypeIndication` value shall equal 0x22 (ISO/IEC 14496-10 Visual Parameter Sets).

4.1.7Elementary Stream Descriptor

If the parameter set elementary stream is present, the elementary stream descriptors for the two streams shall satisfy the following conditions:

(1)The video elementary stream is dependent on the parameter set elementary stream and the `ES_Descriptor` for the video elementary stream shall have a `streamDependenceFlag` equal to true and indicate the ESID of the parameter in the

dependsOnESID field. The streamDependenceFlag in the ES_Descriptor for the parameter set elementary stream shall be false.
(2)The elementary stream clocks for the parameter set elementary stream and the video elementary stream shall be the same and synchronized. The OCRstreamflag and OCR_ES_Id fields in the ESDescriptor for the video elementary stream and parameter set elementary streams shall be used to indicate that both streams share the same OCR.

4.2Codepoints

4.2.1visualProfileLevelIndication

Ed's Note:What should the visual profile level indication in the initial object descriptor be? Is it even necessary to add a new one? The list below does not seem up to date anyway and I do not recall any amendments to MPEG-4 Systems that touch this table.

Replace Table 6 in ISO/IEC 14496 with the following table:
Table 11 - visualProfileLevelIndication Values

Value	Profile	Level
0x00	Reserved for ISO use	-
0x01	Simple	L3
0x02	Simple	L2
0x03	Simple	L1
0x04	Simple Scalable	L2
0x05	Simple Scalable	L1
0x06	Core	L2
0x07	Core	L1
0x08	Main	L4
0x09	Main	L3
0x0A	Main	L2
0x0B	N-Bit	L2
0x0C	Hybrid	L2
0x0D	Hybrid	L1
0x0E	Basic Animated Texture	L2
0x0F	Basic Animated Texture	L1
0x10	Scalable Texture	L3
0x11	Scalable Texture	L2
0x12	Scalable Texture	L1
0x13	Simple Face Animation	L2
0x14	Simple Face Animation	L1
0x15	Advanced Visual Coding	-
0x15-0x7F	reserved for ISO use	-
0x80-0xFD	user private	-
0xFE	no visual profile specified	-

4.2.2objectTypeIndication

Replace Table 8 in ISO/IEC 14496 with the following table:

Table 22 - objectTypeIndication Values

Value	ObjectTypeIndication Description
0x00	Forbidden
0x01	Systems ISO/IEC 14496-1 [a]
0x02	Systems ISO/IEC 14496-1 [b]
0x03-0x1F	reserved for ISO use
0x20	Visual ISO/IEC 14496-2 [c]
0x21	Visual ISO/IEC 14496-10 [e]
0x22	Visual ISO/IEC 14496-10 [e]
0x23-0x3F	reserved for ISO use
0x40	Audio ISO/IEC 14496-3 [d]
0x41-0x5F	reserved for ISO use
0x60	Visual ISO/IEC 13818-2 Simple Profile
0x61	Visual ISO/IEC 13818-2 Main Profile
0x62	Visual ISO/IEC 13818-2 SNR Profile
0x63	Visual ISO/IEC 13818-2 Spatial Profile
0x64	Visual ISO/IEC 13818-2 High Profile
0x65	Visual ISO/IEC 13818-2 422 Profile
0x66	Audio ISO/IEC 13818-7 Main Profile
0x67	Audio ISO/IEC 13818-7 LowComplexity Profile
0x68	Audio ISO/IEC 13818-7 Scaleable Sampling Rate Profile
0x69	Audio ISO/IEC 13818-3
0x6A	Visual ISO/IEC 11172-2
0x6B	Audio ISO/IEC 11172-3
0x6C	Visual ISO/IEC 10918-1
0x6D - 0xBF	reserved for ISO use
0xC0 - 0xFE	user private
0xFF	no object type specified

[a] This object type shall be used for all streamTypes defined in ISO/IEC 14496-1 except IPMP streams.

[b] Includes associated Amendment(s) and Corrigendum(a).

[c] Includes associated Amendment(s) and Corrigendum(a). The actual object types are defined in ISO/IEC 14496-2 and are conveyed in the DecoderSpecificInfo as specified in ISO/IEC 14496-2, Annex K.

[d] Includes associated Amendment(s) and Corrigendum(a). The actual object types are defined in ISO/IEC 14496-3 and are conveyed in the DecoderSpecificInfo as specified in ISO/IEC 14496-3 subpart 1 subclause 6.2.1.

[e] Includes associated Amendment(s) and Corrigendum(a). The actual object types are defined in ISO/IEC 14496-10 and are conveyed in the DecoderSpecificInfo as specified in this admendment, subclause 4.1.5.

4.3Extensibility

The configuration description format may be extended in future versions of this specification. Similarly, the definition of an AVC Access Unit allows for extension by future versions of this specification.

5File Format

5.1Introduction

This subclause defines how to store ISO/IEC 14496-10 data in the MP4 file format. The storage of AVC content uses the existing capabilities of the MP4 file format but also defines new features to support the following features of the AVC codec in the MP4 file format:

Switch Pictures: To enable switching between different coded streams and substitution of pictures within the same stream.

Enhanced Group of Pictures: Provides a structuring of the dependencies of a group of pictures to provide both random access and flexible stream structure (e.g. temporal scalability and layering).

Parameter Sets: The values needed to decode a video coding data are grouped together into a singleparameter set, which is referenced from each slice.

5.2Terminology

AVC	MPEG-4 Systems	MP4
-	Presentation	Movie
Stream	Elementary Stream	Track
Picture	Access Unit	Sample

AVC File type and identification

Conformance with this specification is indicated by the presence of the brand 'jvt1' in the compatible-brands list of the `FileTypeBox` as defined in the ISO Media File Format specification [ISO 14496-1:2001, Amd. 5].
Readers conformant to this specification should read the file if this brand occurs in the compatible-brands list. Other structures and/or track types, defined in other specifications, may be present, and these may be ignored by a conformant reader.
If the major-brand is 'jvt1', then

the minor-version, as defined in this version of this specification, shall be 1.
the file extension shall be ".jvt" [to be checked for collision]
the mime type used to serve the file shall be "video/jvt" [to be allocated]

AVC Track Structure

In the terminology of the ISO Media File Format Specification, an AVC track is a video or visual track. It therefore uses:

a `handler_type` of `'vide'` in the `HandlerBox`;
a video media header `'vmhd'`;

and, as defined below, a derivative of the `VisualSampleEntry`.

Canonical order and restrictions

The AVC elementary stream is stored in the MP4 file format in a *canonical* format. The canonical format is as *neutral* as possible so that systems that need to customize the stream for delivery over different transport protocols – MPEG-2 Systems, RTP, and so on — should not have to *remove* information from the stream while being free to *add* to the stream. Furthermore, a canonical format allows such operations to be performed against a known initial state.

The canonical stream format is an AVC video stream that satisfies the following conditions:

No parameter sets in the video elementary stream. Allparameter set information shall be stored in the sample descriptions for the track. This ensures that it is not necessary to examine every part of the elementary stream to find relevant parameter sets. It also avoids dependencies of indefinite duration between the sample that contains the parameter set definition and the samples that use it. Parameter sets may be inserted into the video elementary stream when the file is streamed over a transport that permits such parameter set updates.

All timing information is external to stream. Timing SEI messages that define presentation or composition timestamps shall not be included in the AVC elementary stream. All timing information is stored in the various sample metadata tables. If timing information were embedded in the stream, then there would be a great danger of inconsistency, and a key feature of MP4, that editing can take place without modifying elementary streams, would break.

All SEI messages are at the beginning of an AU. NAL units included in an access unit shall be ordered so that all SEI messages occur before all video NAL units (slices or data partitions) in the access unit. This allows a rapid inspection of the sample to determine whether it contains SEI messages. All SEI messages shall be contained in a single SEI message NAL unit.

Slice and Partition Ordering. All slices in a single pictureshall occur in scan order. Similarly, all data partitions of a slice shall occur in the order: 'A' partition, 'B' partition, and 'C' partition. No interleaving of partitions from different slices is allowed.

□*No start codes.* The elementary stream shall not include start codes. As stored, each NAL Unit is preceded by a length field; this enables easy scanning of the sample's NAL Units. Systems that use start-codes will reformat the stream to use start codes.

No picture delimiter NAL units. Picture delimiting is defined by the sample structure of the file format. Picture delimiter NAL units shall not be present in the sample data.

No filler data. Video data is naturally represented as VBR in the file format and should be filled for transmission if needed. Filler NAL units and Filler SEI messages shall not be present in the file format stored stream.

Note that these restrictions are more restrictive than the use of AVC in MPEG-4 systems.

5.6Sample format

A file format sample is a picture in AVC terms, and an access unit in MPEG-4 terms. The sample data shall follow the syntax and semantics defined for AVC video access

units in subclause 4.1.4. Note, this format is used for storing data only and does not require the use of MPEG-4 systems framework to carry AVC visual data.

5.7Template Fields Used

The ISO Media File Format defines a number of fields which have default values but which may be defined for use by specific sub-systems. Tracks containing AVC data use the following template fields:
a)`alternate_group` in the `TrackHeaderBox` (see subclause 5.135.13 on stream switching);

5.8Visual Width and Height

The width and height fields in a `VisualSampleEntry` must correctly document the encoded width and height as indicated by the sequence parameter sets in the AVC stream that is described by that entry.
The width and height fields in the track header may not be the same as the width and height fields in the one or more `VisualSampleEntry` in the video track. As specified in the ISO Media File Format,all the sequences are normalized to the track width and height for presentation.

5.9HRD Conformance

[TBD]

5.10Sample Table elements

5.10.1Sample description name and format

5.10.1.1Sample Description Box

5.10.1.1.1Definition

Box Types: 'jvt1', 'jvtC', 'm4ds'
Container: Sample Table Box ('stbl')
Mandatory: Yes
Quantity: Exactly one
An AVC visual sample entry shall contain an AVC Visual Stream Configuration Box, as defined below. This includes a `AVCVideoDecoderConfigurationRecord`, as defined in subclause 4.1.5.1 as the content of `DecoderSpecificInfo` for MPEG-4. Extension descriptors that should be inserted into the Elementary Stream Descriptor, when used in MPEG-4, may also be present.
Multiple sample descriptions may be used, as permitted by the ISO Media File format specification, to indicate sections of video that use different configurations of parameter sets.

5.10.1.1.2Syntax

```
    // Visual Sequences
class AVCConfigurationBox extends Box('jvtC') {
    AVCVideoConfigurationRecord() AVCConfig;
}

class MPEG4DescriptorsBox extends Box('m4ds') {
    Descriptor Descr[0 .. 255];
}

class AVCSampleEntry() extends VisualSampleEntry ('jvt1'){
    AVCConfigurationBox  config;
    MPEG4ExtensionDescriptorsBox (); // optional
}
```

5.10.1.1.3Semantics

`Compressorname` in the base class `VisualSampleEntry` indicates the name of the compressor used with the value `"\016JVT/AVC Coding"` being recommended (\016 is 14, the length of the string as a byte)
`config` is defined in subclause 4.1.5.2.
`Descr` is a descriptor which should be placed in the `ElementaryStreamDescriptor` when this stream is used in an MPEG-4 context. This does not include `SLConfigDescriptor` or `DecoderConfigDescriptor`, but includes the other descriptors in order to be placed after the `SLConfigDescriptor`. The use of this field is specified in subclause 5.15.15.15.1.

5.10.1.2Parameter Sets

Each AVC sample description, which contains the AVC video stream decoder specific information, includes a group of sequence and picture parameter sets. This group of parameter sets functions much like a codebook. Each parameter set has an identifier, and each slice references the parameter set it was coded against using the parameter set's identifier.
In the file format each configuration of parameter sets is represented separately. A parameter set cannot be updated without causing a different sample description to be used. For example, suppose that samples 1-4 use picture parameter sets identified as 1, 2, 3 and a single sequence parameter set identified as 1. At sample 5 a new value of picture parameter set 2 is required but picture parameter sets 1 and 3 remain unaltered and are used until sample 10. In this case, the sample description for samples 1 through 4 is the same and contains the initial values of picture parameter sets 1, 2, 3 and sequence parameter set 1. At sample 5 the sample description must change to a second sample description, which contains the updated value for picture parameter set 2 *as well as* the original values of picture parameter sets 1 and 3 and sequence parameter set 1. This second sample description is used for samples 5 through 10.
Systems wishing to sendparameter set updates will need to compare the two configurations to find the differences in order to send the appropriate parameter set updates.

5.10.2Decoding Time (DTS) and Composition Time (DTS)

Samples are stored in the file format in decoding order. If frame reordering is not used, this is also presentation order, and only the time-to-sample 'stts' table is used. Note that any kind of picture may be reordered in AVC video, not only B-frames.
If decoding time must be stored, the composition time-to-sample 'ctts' is also used.

5.10.3Sync sample (IDR)

A sync sample shall correspond to an instantaneously decodable picture in the AVC stream (also known as an Instantaneously Decodable Refresh point (IDR)).

5.10.4Shadow sync

The use of the shadow sync table to indicate alternate encodings of a sample for random access are supported for as defined in the ISO Base Media File format. A shadow sync shall indicate a sample that is an independently decodable sample, which shall be encoded as either an I-frame or a SI-frame.
While the use of shadow sync for AVC content is supported for backward compatibility reasons, this use is deprecated and use of the mechanisms defined in subclause 5.135.13 is recommended.

5.11Sample Groups

5.11.1Introduction

This subclause specifies a generic mechanism for representing a partition of the samples in a track. A *sample grouping* is an assignment of each sample in a track to be a member of one *sample group,* based on a grouping criterion. A sample group in a sample grouping is not limited to being contiguous samples and may contain non-adjacent samples. As there may be more than one sample grouping for the samples in track, each sample grouping has a type field to indicate the type of grouping. For example, a file might contain two sample groupings for the same track: one based on an assignment of sample to layers and another to sub-sequences.
Sample groupings are represented by two linked data structures: (1) a `SampleToGroup` box represents the assignment of samples to sample groups; (2) a `SampleGroupDescription` box contains a *sample group entry* for each sample group describing the properties of the group. There may be multiple instances of the `SampleToGroup` and `SampleGroupDescription` boxes based on different grouping criteria. These are distinguished by a type field used to indicate the type of grouping.
One example of using these tables is to represent the assignments of samples to *layers*. In this case each sample group represents one layer, with an instance of the `SampleToGroup` box describing which layer a sample belongs to. For more details, please refer to subclause 5.125.12

5.11.1.1SampleToGroup Box

5.11.1.1.1Definition

Box Type: 'sbgp'
Container: Sample Table Box ('stbl')
Mandatory: No
Quantity: Zero or more.

This table can be used to find the group that a sample belongs to and the associated description of that sample group. The table is compactly coded with each entry giving the index of the first sample of a run of samples with the same sample group descriptor. The sample group description ID is an index that refers to a `SampleGroupDescription` box, which contains entries describing the characteristics of each sample group.

There may be mutiple instance of this box if there is more the one sample grouping for the samples in a track. Each instance of the `SampleToGroup` box has a type code that distinguishes different sample groupings. Within a track, there shall be a most instance of this box with a particular grouping type. The associated `SampleGroupDescription` shall indicate the same value for the grouping type.

5.11.1.1.2Syntax

```
aligned(8) class SampleToGroupBox
                                                    extends FullBox('sbst',
version = 0, 0)
{
                                                    unsigned int(32)
                                                    grouping_type;
                                                    unsigned int(32)
                                                    entry_count;
                                                    for (i=1; i <= entry_count;
i++)
                                                    {
                                                     unsigned int(32)
                                                    sample_count;
                                                     unsigned int(32)
                                                    group_description_index;
                                                    }
}
```

5.11.1.1.3Semantics

`version` is an integer that specifies the version of this box.

`grouping_type` is an integer that identifies the type (i.e. criterion used to form the sample groups) of the sample grouping and links it to its sample group description table with the same value for grouping type. At most once occurrence of this box with the same value for `grouping_type` value shall exist for a track.

`entry_count` is an integer that gives the number of entries in the following table.

`sample_count` is an integer that gives the number of consecutive samples in the group.

`group_description_index` is an integer that gives the index of the sample group entry which describes the samples in this group. The index ranges from 1 to the number of sample group entries in the `SampleGroupDescription` Box.

5.11.1.2Sample Group Description Box

5.11.1.2.1Definition

Box Types: 'sgpd'
Container: Sample Table Box ('stbl')
Mandatory: No
Quantity: Zero or more, with one for each Sample Group Table box.
This description table gives information about the characteristics of sample groups. The descriptive information is any other information needed to define or characterize the sample group.
There may be mutiple instance of this box if there is more the one sample grouping for the samples in a track. Each instance of the `SampleGroupDescription` box has a type code that distinguishes different sample groupings. Within a track, there shall be a most instance of this box with a particular grouping type. The associated `SampleToGroup` shall indicate the same value for the grouping type.
The information is stored in the sample group description box after the entry-count. An abstract entry type is defined and sample groupings shall define derived types to represent the description of each sample group. For video tracks, an abstract `VisualSequenceEntry` is used with similar types for audio and hint tracks.

5.11.1.2.2Syntax

```
// Sequence Entry
abstract class SampleGroupDesriptionEntry (unsigned int(32) handler_type)
{
}

// Visual Sequence
abstract class VisualSampleGroupEntry (type) extends SampleGroupDesriptionEntry
(type)
{
}

// Audio Sequences
abstract class AudioSampleGroupEntry (type) extends
SampleGroupDesriptionEntry(type)
{
}

aligned(8) class SampleGroupDescriptionBox (unsigned int(32) handler_type)
                                                    extends FullBox('sgpd', 0,
0){
                                                    unsigned int(16)
```

```
grouping_type;                                          unsigned int(32)

entry_count;                                            int i;
                                                        for (i = 1 ; i <= entry_count
; i++){
                                                         switch (handler_type){
                                                              case 'vide': // for video
tracks

                                                        VisualSampleGroupEntry
();
                                                                 break;
                                                              case 'soun': // for
audio tracks

                                                        AudioSampleGroupEntry();
                                                                 break;
                                                              case 'hint': // for hint
tracks

                                                        HintSampleGroupEntry();
                                                                 break;
                                                         }
                                                        }
}
```

5.11.1.2.3Semantics

`version` is an integer that specifies the version of this box.
`grouping_type` is an integer that identifies the `SampleToGroup` box that is associated with this sample group description.
`entry_count` is an integer that gives the number of entries in the following table.

5.12Layering and sub-sequences

Streams may be constructed so that the referential dependencies between samples allow only subsets of the samples to be sent to the decoder. This mechanism is called *thinning* a stream and is entirely transparent to the decoder receiving the elementary stream. Thinning discards entire sets of samples using knowledge of what other sets of pictures this set of pictures depends on and what picture sets in turn depend on it.
The referential dependencies between samples in a stream are structured into *layers* and *sub-sequences*. Samples in higher layers can only depend on samples in lower layers. Layers are numbered, and the samples organized such that a sample in layer N has no dependencies on samples in layers greater than N.

Sub-sequences are sets of samples that have external referential dependencies on other sets of samples. Dependency relations between sub-sequences represent the dependency structure of a stream. The samples in a sub-sequence can be disposed of as a unit when thinning streams. Each sub-sequence belongs to one and only one layer. A sample shall reside in one layer and in one sub-sequence only.
Layering and sub-sequences are not features of the codec but this information is represented in the file format to allow systems reading the files to understand the ways in which stream thinning may be achieved without having to examine the dependency structure of every sample.

5.12.1.1Sub-Sequence Description Entry

Box Types: 'jvss''
Container: Sample Group Description Box ('sgpd')
Mandatory: No
Quantity: Zero or more.
A sub-sequence description entry is a sample group entry that describes a sub-sequence. A sub-sequence is a set of samples in a track belonging to the same layer. A sub-sequence depends on another sub-sequence if and only if there exists a sample in the this sub-sequence that is directly referentially dependent on some sample in the other sub-sequence. All sample in a sub-sequence shall be depend only on other samples within the same sub-sequence or samples in the sub-sequences on which is it dependent. A sub-sequence can depend on zero or more sub-sequences in the same or lower layers. A sub-sequence shall not depend on sub-sequences in the higher scalability layers.

5.12.1.1.1Syntax

```
aligned(8) class DependencyInfo
{
                                                    unsigned int(8)
layerNumber;
                                                    unsigned int(32)
subSequenceIdentifier;
}

class AVCSubSequenceEntry () extends VisualSampleGroupEntry ('jvss')
{
                                                    unsigned int(32)
subSequenceIdentifer;
                                                    unsigned int(32)
avgBitRate;
                                                    unsigned int(32)
avgFrameRate;
                                                    unsigned int(16)
numReferences;
                                                    DependencyInfo
dependency[numReferences];
```

}

}

5.12.1.1.2Semantics

`layerNumber` and `subSequenceIdentifier` within the `DependencyInfo` class identify a sub-sequence that is used as a reference for this sub-sequence.
`subSequenceIdentifier` in the Sub-Sequence Box gives the identifier for the sub-sequence.
`avgBitRate` gives the average bit rate in bits/second of the sub-sequence within the segment. All NAL units belonging to pictures in the layer are taken into account in the calculation. A value of zero means an undefined bit rate.
`avgFrameRate` gives the average frame rate in frames/(256 seconds) of the sub-sequence within the segment. A value of zero means an undefined frame rate.
`numReferences` gives the number of sub-sequences directly referenced in this sub-sequence. `dependency` is an array of `DependencyInfo` structures giving the identification information of the referenced sub-sequences.

5.12.1.2Layer Description Entry

Box Types: 'jvll'
Container: Sample Group Description Box ('sgpd')
Mandatory: No
Quantity: Zero or more.
A layer sample group entry defines the layer information for all samples in a layer. Layers are ordered hierarchically based on their dependency on each other: The first or base layer is independently decodable. The second and subsequent layers are enhancement layers that depend on some data in the base layer. Pictures in a higher layer can only depend on pictures in lower layers.

5.12.1.2.1Syntax

```
class AVCLayerEntry() extends VisualSampleGroupEntry ('jvll')
{
                                                    unsigned int(8)
layerNumber;
                                                    unsigned int(32)
avgBitRate;
                                                    unsigned int(32)
avgFrameRate;
}
```

5.12.1.2.2Semantics

`layerNumber` gives the number of this layer with the base layer being numbered as zero and all enhancement layers being numbered as one or higher with consecutive numbers.

`avgBitRate` gives the average bit rate in bits/second of the layer within the segment. A value of zero indicates an undefined bit rate.
`avgFrameRate` gives the average frame rate in frames per 1000 seconds of this layer with a value of zero indicating an undefined frame rate.

5.13Alternate Streams and Switch Pictures

In typical streaming scenarios, one of the key requirements is to scale the bit rate of the compressed data in response to changing network conditions. The simplest way to achieve this is to encode multiple streams with different bandwidths and quality settings for representative network conditions. The server can then switch amongst these pre-coded streams in response to network conditions. In current standards, switching between streams is only possible at I-frames, because the frames can only be switched when there are no dependencies on prior pictures for reconstruction.
AVC has adopted SP-pictures and SI-pictures ("switching pictures") that allow switching from one stream to another while still supporting inter-frame coding of switch pictures. Figure 3 shows how SP frames are used to switch between two different bit streams.

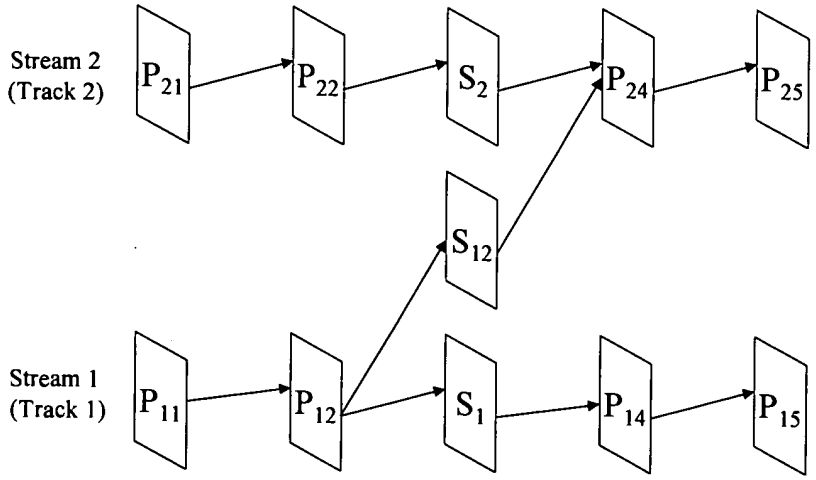


Figure 3: **Stream Switching**

In the file format, switch pictures are stored in *switch picture* tracks, which are tracks separate from the track that is being switched from and the track being switched to. Switch picture tracks can be identified by the existence of a specific required track reference in that track. A switch picture is an alternative to the sample in the destination track that has exactly the same decoding time. If all switch pictures are SI pictures, then no further information is needed.
If any of the pictures in the switch track are SP pictures, then two extra pieces of information may be needed. First, the source track, that is being switched from, must be identified by using a track reference (the source track may be the same track as the destination track). Second, the dependency of the switch picture on the samples in the source track may be needed, so that a switch picture is only used when the pictures on which it depends have been supplied to the decoder.
This dependency is represented by means of an optional extra sample table. There is one entry per sample in the switch track. Each entry records the relative sample numbers in the source track on which the switch picture depends. If this array is empty for a given

sample, then that switch sample contains an SI picture. If the dependency box is not present, then only SI-frames shall be present in the track.

5.13.1Alternate Group

The ISO media file format (but not the version one specification of the MPEG-4 file format, which is branded as 'mp41') supports the use of what is called *alternate tracks*. Each track can optionally specify an *alternate group* (in the track header box) that groups together alternate encodings of the same content. Thus, each alternate bit-stream can be stored as a separate track in the MP4 and related together as alternate tracks. All the tracks which form a group which may be switched between, but not the tracks containing the switch pictures, must be a member of an `alternate_group` with a non-zero group identifier.
An alternate group is not needed if there is only one primary track, with a switch track. This switch track may contain SI pictures, or SP pictures for trick modes or error resilience, which predict both from and to the same track.

5.13.2Track references

The switch track must be linked to the track into which it switches (the destination track) by a track reference of type 'swto' in the switch-picture track.
If the switch track contains SP pictures, the switch track must be linked to the track from which it switches (the source track) by a track reference of type 'swfr' in the switch-picture track.

5.13.3Sample dependency

This subclause defines the dependencies of each switch sample on sample(s) in the source track. This table is only needed in a switch track which has a source ('swfr') track dependency.

5.13.3.1Definition

Box Type: 'sdep'
Container: Sample Table 'stbl'
Mandatory: No
Quantity: Zero or exactly one.
This box contains the sample dependencies for each switch sample. The dependencies are stored in the table, one record for each sample. The size of the table, `sample_count` is taken from the `sample_count` in the Sample Size Box ('stsz').

5.13.3.1.1Syntax

```
aligned(8) class SampleDependencyBox
    extends FullBox('sdep', version = 0, 0) {
    int i;
    for (i=0; i < sample_count; i++) {
        unsigned int(16) dependency_count;
        for (j=0; j < dependency_count; j++) {
            signed int(16) relative_sample_number;
```

```
        }
    }
}
```

5.13.3.1.2Semantics

`dependency_count` - is an integer that a counts the number of samples in the source track on which this switch sampledepends.
`relative_sample_number` - is an integer that identifies a sample in the source track. The relative sample numbers are encoded as follows. If there is a sample in the source track with the same decoding time, it has a relative sample number of 0. Whether or not this sample exists, the sample in the source track which immediately precedes the decoding time of the switch sample has relative sample number –1, the sample before that –2, and so on. Similarly, the sample in the source track which immediately follows the decoding time of the switch sample has relative sample number +1, the sample after that +2, and so on.

5.14Representation of new structures in Movie Fragments

[TBD]

5.15Use of the AVC File Format in MPEG-4 Systems

This subclause specifies how the AVC file format shall be used when the file is marked as being compatible with the MPEG-4 file format specified in ISO/IEC 14496-1:2001 Amendment 6. This subclause applies when the file is branded with the MPEG-4 file format brand of 'mp41' or 'mp42', and the AVC video data must be used in an MPEG-4 systems context.

5.15.1Forming the Elementary Stream Descriptor

As is normal for MPEG-4 streams, the TrackID is related to the `ElementaryStreamID`, and the `SLConfigDescriptor` is generated following the rules for any MPEG-4 stream The format of the ES Descriptor is specified in chapter 13 of the MPEG-4 systems specification [ISO/IEC 14966:2001].
If the Elementary Stream Descriptor should contain any other descriptors than `SLConfigDescriptor` or `DecoderConfigDescriptor`, they are stored in the Sample Description as defined in subclause 5.10.1.15.10.1.1.

5.15.2Forming the DecoderConfigDescriptor

The object type and stream type indications have the values defined in subclause 4.2 for AVC streams.
The `buffersizeDB`, `maxBitrate`, and `avgBitrate` fields can be filled by inspection of the VUI Sequence Parameters, if present in the sequence parameter set for the AVC stream.

The `DecoderSpecificInfo` is formed using the contents of the `AVCConfigurationBox`.

5.15.3Profile and Level Indication

The MPEG-4 `ProfileLevelIndication` can be deduced from the AVC Profile and Level Indication found in the `AVCDecoderConfiguration`.
[Editor's note: Until the code-points are assigned, it's not clear whether this is a one-one mapping.]

5.15.4Elementary Stream Descriptors

The use of multiple non-dependent elementary stream descriptors may *also* be indicated by the presence of more than one independent AVC video elementary stream descriptors in an object descriptor.

5.15.5Switch Picture Tracks

Switch picture tracks are *not* MPEG-4 elementary streams and shall not be included within an MPEG-4 object descriptor.

5.16Possible issues

5.16.1Need a sample fragment table?

Is the use of in-line size indicators for NAL Units sufficient or should those sizes be represented in new meta-data tables? See, for reference, the proposed sample fragment metadata tables in N8708.

5.16.2Typical patterns of NAL Unit sizes etc.

This design could probably benefit from more knowledge of typical patterns of use and sizes etc. For example, repeating patterns etc. can be leveraged to reduce overhead, and the sizes of fields adjusted (e.g. the relative sample number). See, for example, N8708.

5.16.3Do we need better visibility into where SEIs occur?

If an application wishes to answer "where are the SEI messages in this stream?" it must search all samples for those beginning with SEIs. We are unaware of a need for such a search, but if such need arises, some kind of indexing for the SEIs may be needed.

5.16.4Alternatives are tracks but layers are in-stream?

Alternative designs for switching could put the streams into the same track, or for layering could put the layers into different tracks.

5.16.5Do we need to allow parameter set updates in the OD stream?

The file format effectively allowing update of `DecoderSpecificInfo`, which might change parameters such as the visual width and height (though the file format normalizes these). Does MPEG-4 systems need to allow such re-configuration of existing streams,

or should we express constraints on the changes in configuration permitted in an MPEG-4 context?

5.16.6Will chunking be affected by Parameter Set changes?

All the samples in a chunk share a single configuration. If parameter sets change frequently, and this causes the configuration to change frequently, this would have a significant adverse impact on the size of the file-format sample tables. Typical patterns of use would again be of assistance here.

5.16.7SEI Messages and Parameter Sets

Should access units without any video data and containing only SEI messages and/or parameter sets be allowed? Currently, such access units are not supported.

5.16.8Non-stored pictures

When decoding frames to 'roll forward' or other random access, it would save a lot of disk accesses if it were evident from the sample table what was a non-stored frame (i.e. frames on which no others depend).

5.16.9Picture Delimiters

Should we always have a picture delimiter in the frames? If a picture delimiter is present, perhaps it ought to be first, before the SEIs?

5.16.10Timing of Parameter Set Updates in Parameter Set Streams

Should parameter set streams provide for parameter set NAL units whose DTS is *equal* to the DTS time of the first NAL unit (in the video elementary stream) that refers to that parameter set? Or should the DTS of a parameter set NAL unit (in the parameter set elementary stream) always be *prior* to the decoding time of the first NAL unit (in the video elementary stream) that refers to the parameter set?
Note: In the above the DTS of an NAL unit means the DTS of the access unit containing the NAL unit.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

SUB-PICTURE: ROI CODING AND UNEQUAL ERROR PROTECTION

Miska M. Hannuksela, Ye-Kui Wang**, and Moncef Gabbouj****

*Nokia Mobile Software, Tampere, Finland
**Tampere International Center for Signal Processing (TICSP), Tampere, Finland
*** Tampere University of Technology, Finland

ABSTRACT

Region-of-interest coding and unequal error protection are two important tools in video communication systems to improve received visual quality. One common property of the two techniques is that unequal coding or transmission is applied to improve the quality of the most important parts of images. The proposed sub-picture coding technique facilitates both region-of-interest coding and unequal error protection by partitioning images to regions of interest and separating the corresponding coded data units from each other. Simulation results show that the overall subjective quality is considerably improved compared to the conventional coding schemes.

1. INTRODUCTION

1.1. Slice-Based Coding

Modern video coding standards, such as ITU-T Recommendation H.263 and ISO/IEC MPEG-4 Part 2, allow division of coded pictures to slices. Slices can be regarded as a way to split a bit-stream to transport packets that can be decoded independently. While spatial and syntactical prediction is disabled across slice boundaries, motion vectors may cross slice edges. This fact causes spatio-temporal error propagation, when motion vectors point to areas that are reconstructed incorrectly. In order to prevent this phenomenon, H.263 includes the optional independent segment decoding mode (H.263 Annex R). When this optional mode is in use, slice boundaries are treated as picture boundaries, and therefore no spatio-temporal error propagation over slice boundaries occurs. Due to restricted motion prediction, compression efficiency drops compared to normal slice-based operation.

1.2. Region-Of-Interest Coding

Region-of-interest (ROI) coding refers to techniques that code some spatial pictorial regions using more refined method than the rest. One method is to apply smaller quantization parameters (QP) hence allocate more bits to the region of interest [1]. As a result, better quality can be observed in the interested region; and hence the overall subjective quality is improved. A shortcoming of ROI coding is that it may cause a boundary effect between the ROI region and the leftover region due to unequal coding.

1.3. Unequal Error Protection

Unequal error protection (UEP) refers to techniques that protect part of the transmitted bit-stream in the transport system better than the rest. Examples of applicable UEP techniques include application-layer selective retransmission [2][3], transport-layer forward error control (e.g. RFC 2733 [4]), guaranteed network Quality of Service (e.g. QoS architecture of Universal Mobile Telecommunication System [5]), and Differentiated Services (DiffServ) [6][7].

In order to apply unequal error protection, video bit-streams have to be organized in portions of different importance in terms of visual quality. Techniques achieving this goal include data partitioning and scalable video coding. Data partitioning refers to a technique where subjectively equally important codewords of all macroblocks (MBs) in a slice are partitioned into a continuous block of data. Typically, MB headers and motion information form one partition and coded prediction error blocks form another partition. Scalability refers to the capability of a compressed sequence to be decoded at different bit-rates. Scalability can be further categorized into temporal, SNR, and spatial scalability. Video coding schemes with temporal scalability is reviewed in [8], spatial and SNR scalable video coding are reviewed in [9].

1.4. Overview of the Paper

Conventional ROI coding treats the entire bit-stream equally in transmission, while data partitioning and scalable coding generally treat coding of an entire image equally in spatial domain. Arbitrarily shaped objects [10], as defined in the MPEG-4 standard, can be used to extract the regions of interest. However, the arbitrary shape

coding often inherits relatively high processing requirements.

This paper proposes a simple region-based coding method, called sub-picture coding, which is based on rectangular shapes. It can be considered as a simplification of arbitrary shape coding. While sub-picture coding does not provide means to define object boundaries accurately, it suits simultaneously ROI coding and unequal error protection. In addition, a gradual ROI bit allocation method is proposed to reduce the boundary effect between the selected ROI region and the leftover region.

Section 2 of the paper presents the sub-picture coding method. The gradual ROI bit allocation method is described in Section 3. Simulation results of the proposed technique are provided in Section 4. The paper is concluded in Section 5.

2. SUB-PICTURE CODING

The proposal adds a sub-picture coding layer between picture and slice layers. Sub-pictures are rectangular except for the so-called background sub-picture, which consists of the picture area not falling to any of the rectangular sub-pictures. Rectangular sub-pictures are also referred to as foreground sub-pictures. Sub-pictures boundaries are aligned with macroblock (MB) boundaries. Sub-pictures do not overlap. A slice resides within one sub-picture only. A slice in a background sub-picture may not contain spatially adjacent MBs within a MB line, as it can be intervened by foreground sub-pictures. Figure 1 shows an example of a picture having one foreground sub-picture and one background sub-picture, both of which include two slices.

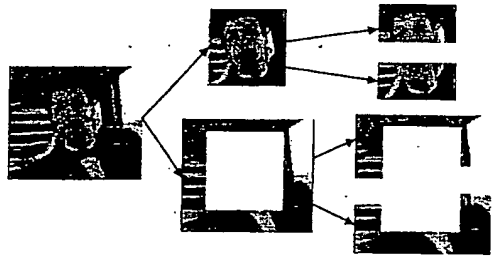


Fig. 1. Example of picture, sub-pictures, and slices.

There are two coding modes associated with sub-pictures: normal mode and independent mode. In the normal mode, sub-picture boundaries are treated as slice boundaries. In the independent mode, boundaries of foreground sub-pictures are treated as picture boundaries and sub-picture segmentation is static over a group of pictures. In other words, in the independent mode, temporal and spatial prediction over sub-picture boundaries is prevented when coding the foreground sub-pictures. No such limitation exists when coding a background sub-picture.

Coding of the foreground sub-pictures takes place before coding of the background sub-picture, therefore different coding methods can applied conveniently to different regions, and the bit-streams of different regions are separated to suit UEP.

3. GRADUAL ROI BIT ALLOCATION

Conventional ROI bit allocation methods, such as the methods proposed in [1], assign two QP values, generally with a large difference, for each picture, one for the ROI region and the other for leftover region. As a result, annoying boundary effects are caused between the two regions. To reduce the boundary effects, a gradual ROI bit allocation scheme described in the following is proposed.

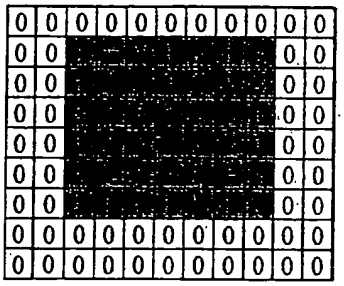


Fig. 2. Gradual ROI bit allocation/quantization.

The background QP, QPb, is fixed for all background MBs. The foreground QP is a function of the distance from regional edge: if a MB is adjacent to the edge, its QP is QPb-1; if the distance between a MB and the edge is larger (e.g. one MB), the QP of the MB is QPb-2; for other foreground MBs, the QP is QPb-3. An example is shown in Figure 2, where the gray colored MBs form the foreground sub-picture.

4. SIMULATIONS

4.1. Conditions

We selected multicast Internet streaming as a target application. As interactive error concealment cannot be used in large scale with IP multicast, transport coding level forward error correction (FEC) according to RFC 2733 [4] was used. We used the ITU-T Video Coding Experts Group (VCEG) common conditions for the low-delay Internet applications [11] as much as we considered appropriate.

Sub-picture coding and the gradual ROI bit allocation were implemented in and tested with H.26L Test Model Long-Term (TML) version 8.6 [12][13]. A constant foreground sub-picture was selected, and finer QPs

decided by the gradual ROI bit allocation method were used in the foreground region. The independent sub-picture decoding mode was in use. The scheme was compared to TML-8.6 with and without ROI quantization (abbreviated as Conv+ROI and Conv, respectively). The ROI quantization for the conventional codec was the same as for the sub-picture coding scheme.

The simulator consisted of the following phases. First, the TML encoder was used to encode a sequence and generated an RTP packet stream. Then, a packet loss simulator erased some of the generated packets according to the error patterns released in [14]. The resulting packets stream was decoded using the TML decoder.

Encapsulation into RTP packets was done as follows: In the sub-picture coding scheme, INTRA pictures were encapsulated into five packets. There were two packets for the foreground sub-picture: one packet contained odd MB rows and another packet contained even MB rows. This slice interleaving mechanism, introduced in [15], was used to obtain a better error concealment result. One parity FEC packet was generated for the two foreground packets according to RFC 2733. The background sub-picture was packetized into another two packets using slice interleaving method. Two consecutive INTER pictures consisted a group, and for each such group there were two foreground sub-picture packets, one parity FEC packet for the foreground packets, and two background sub-picture packets. A sub-picture packet contained data from two pictures: MBs from even rows of a certain frame and MBs from odd rows of the next frame or vice versa. When sub-picture coding was not in use, there were three packets for each INTRA and INTER frame: two packets for the entire picture (slice interleaving applied), and one parity FEC packet for the two packets.

As multicast Internet streaming was assumed, the coding and transport scheme was tailored for the worst expected case. We selected optimal coding parameters and encapsulation options by trial and error according to the PSNR performance in the 20 % packet loss rate. We tested several INTRA macroblock coding rates as well as several slice sizes to find the optimal ones. The packet stream obtaining the best PSNR performance was multicast. The PSNR performance in 0, 3, 5, and 10 % packet loss rate was then simulated with the multicast packet stream. In order to obtain statistically significant results, each packet stream was virtually transported ten times for each packet loss rate. The first loss pattern position of a run starts from where the previous run ended.

4.2. Results

The experiments were done using Carphone, Hall, Coastguard, Foreman, News and Irene sequences, with different frame rates and bit-rates. We present only part of the results due to lack of space, more results can be obtained from the authors.

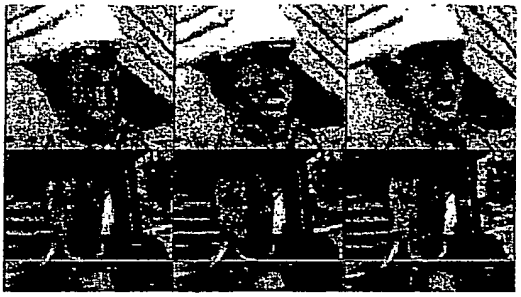


Figure 3. Example snapshots, 20 % packet loss rate. From left to right, the used codecs are Conv, Conv+ROI, and the proposed codec.

Figure 3 shows some example snapshots of Foreman@64 kbps and Carphone@64kbps in 20 % packet loss rate. From left to right in the figure, the snapshots were produced by the Conv, Conv+ROI, and the proposed codec, respectively. It can be seen that in both sequences the proposed sub-picture coding scheme with gradual bit allocation maintains the best subjective image quality.

Figure 4 presents the average luminance PSNR values of the foreground area of Irene@384kbps, News@64kbps, Foreman@64kbps, and Carphone@64kbps. These objective results confirm the subjective ones. Sub-picture coding improves the average luminance PSNR of the foreground area consistently regardless of the packet loss rate. This improvement is gained at the expense of background image quality, which is degraded by coarser quantization and less error protection. In fact, the overall PSNR in the sub-picture coding case drops a little compared to the ROI quantization case and somewhat more compared to the constant quantization case. However, errors in the background are far less noticeable than errors in the foreground, and therefore the overall subjective quality is improved.

5. CONCLUSION

Many video communication systems can make use of region-of-interest coding and unequal error protection to improve received visual quality. In order to apply them, pictures should be partitioned to different regions, or video bit-streams have to be organized in portions, of different importance in terms of visual quality. Techniques achieving these goals include data partitioning, scalable video coding, and region-based video coding.

This paper introduced a simple region-based video coding method called sub-picture coding. It provides means for both region-of-interest coding and unequal error protection based on partitioned regions of interest. In sub-picture coding, a picture is partitioned to one or more

BEST AVAILABLE COPY

rectangular foreground sub-pictures and to a hollow background sub-picture along macroblock boundaries. Sub-picture edges are treated as slice boundaries or as picture boundaries depending on the signaled coding mode.

The sub-picture coding method was simulated in multicast Internet streaming conditions and it was found to be superior to conventional coding. Due to its simplicity and the obtained results, we think that sub-picture coding has some potential to become one of the key video coding tools for region-of-interest coding and unequal error protection.

6. REFERENCES

[1] D. Chai, K. N. Ngan, and A. Bouzerdoum, "Foreground / Background Bit Allocation for Region-Of-Interest Coding", IEEE 2000 International Conference on Image Processing, vol. 2, pp. 923-926, 2000.

[2] A. Miyazaki, H. Fukushima, K. Hata, T. Wiebke, R. Hakenberg, C. Burmeister, N. Takatori, S. Okumura, and T. Ohno, "RTP Retransmission Payload Format", IETF Internet Draft draft-ietf-avt-rtp-selret-02.txt, 18 July 2001.

[3] D. Leon, V. Varsa, "RTP retransmission framework", IETF Internet Draft draft-leon-rtp-retransmission-00.txt, July 2001.

[4] J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction" IETF RFC 2733, December 1999.

[5] 3GPP TS 23.107, "QoS Concept and Architecture (Release 4)", V4.2.0, October 2001, available via ftp://ftp.3gpp.org/Specs/2001-09/Rel-4/23_series/23107-420.zip.

[6] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", IETF RFC 2474, December 1998.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services", IETF RFC 2475, December 1998.

[8] M. M. Hannuksela, "Simple Packet Loss Recovery Method for Video Streaming", International Packet Video Workshop PV2001, Kyongju, South Korea, 30 April – 1 May 2001.

[9] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 3, March 2001.

[10] N. Brady, "MPEG-4 Standardized Methods for the Compression of Arbitrarily Shaped Video Objects", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 8, December 1999.

[11] S. Wenger, "Common conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing", ITU-T Video Coding Experts Group, document VCEG-N79r1, September 2001.

[12] T. Wiegand (editor), "H.26L Test Model Long-Term Number 9 (TML-9) draft0", ITU-T Video Coding Experts Group, document VCEG-N83d1, December 2001.

[13] H.26L Test Model Software, version 8.6, October 2001.

[14] S. Wenger, "Error patterns for Internet video experiments", ITU-T SG16 document Q15-I-16-R1, October 1999.

[15] S. Wenger and G. Côté, "Using RFC2429 and H.263+ at low to medium bit-rates for low-latency applications", International Packet Video Workshop PV99, April 1999.

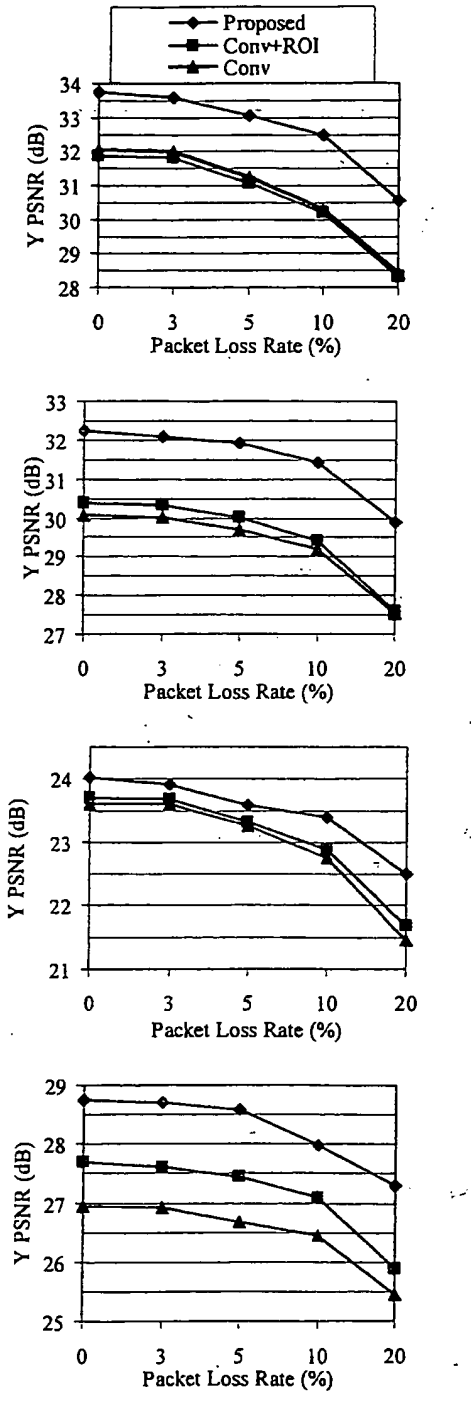


Fig. 4. Foreground average luminance PSNR values of Irene, News, Foreman, and Carphone (up to down)

SUB-PICTURE VIDEO CODING FOR UNEQUAL ERROR PROTECTION

Miska M. Hannuksela, Ye-Kui Wang**, and Moncef Gabbouj****

*Nokia Mobile Phones, Tampere, Finland
**Tampere International Center for Signal Processing (TICSP), Tampere, Finland
*** Tampere University of Technology, Finland

ABSTRACT

Unequal error protection is one of the key tools in video communication systems operating over error-prone networks. In order to allow unequal protection of a video bit-stream, codewords have to be categorized according to their importance to visual quality. The proposed sub-picture coding method allows partitioning images to regions of interest and helps to maintain a good image quality in the chosen regions. As an example, the sub-picture coding scheme is applied to multicast Internet streaming. It is shown that the overall subjective image quality and the objective foreground image quality are considerably better when compared to the selected conventional coding schemes.

1. INTRODUCTION

1.1. Slice-Based Coding

Modern video coding standards, such as ITU-T Recommendation H.263 and ISO/IEC MPEG-4 Part 2, allow division of coded pictures to slices. A slice consists of a number of consecutive macroblocks in raster scan order. Slices can be regarded as a way to split a bit-stream to transport packets that can be decoded independently. While spatial and syntactical prediction is disabled across slice boundaries, motion vectors may cross slice edges. This fact causes spatio-temporal error propagation, when motion vectors point to areas that are reconstructed incorrectly. In order to prevent this phenomenon, H.263 includes the optional independent segment decoding mode (H.263 Annex R). When this optional mode is in use, slice boundaries are treated as picture boundaries, and therefore no spatio-temporal error propagation over slice boundaries occurs. Due to restricted motion prediction, compression efficiency drops compared to normal slice-based operation. Furthermore, in order to avoid prediction confusions in slice boundaries, each slice must reside within one row of macroblocks, and its shape and position must be constant throughout the group of pictures.

1.2. Unequal Error Protection

Term unequal error protection (UEP) refers to techniques that protect part of the transmitted bit-stream in the transport system better than the rest. Examples of applicable UEP techniques include application-layer selective retransmission [1][2], transport-layer forward error control (e.g. RFC 2733 [3]), guaranteed network Quality of Service (e.g. QoS architecture of Universal Mobile Telecommunication System [4]), and Differentiated Services (DiffServ) [5][6].

In order to apply unequal error protection, video bit-streams have to be organized in portions of different importance in terms of visual quality. Techniques achieving this goal include data partitioning and scalable video coding.

Data partitioning refers to a technique where subjectively equally important codewords of all macroblocks in a slice are partitioned into a continuous block of data. Typically, macroblock headers and motion information form one partition and coded prediction error blocks form another partition.

Scalability refers to the capability of a compressed sequence to be decoded at different bit-rates. Scalability can be further categorized into temporal, SNR, and spatial scalability. Video coding schemes with temporal scalability have been reviewed in [7]. A spatially or SNR-scalable bit-stream [8] is organized into a base layer and one or more enhancement layers, and each additional enhancement layer improves perceptual quality. Fine granularity scalability (FGS) [8] is a specific type of SNR scalability. The FGS coding technique arranges the bit-stream into a base layer and an enhancement layer. Each coded picture in the enhancement layer can be truncated into any number of bits, and the number of enhancement layer bits is proportional to the image quality.

Data partitioning and scalable coding techniques generally treat an entire image equally in spatial domain. However, many images have distinct spatial regions of interest. These regions could have better error protection than other areas in order to obtain a better subjective quality compared to coding and transport schemes that

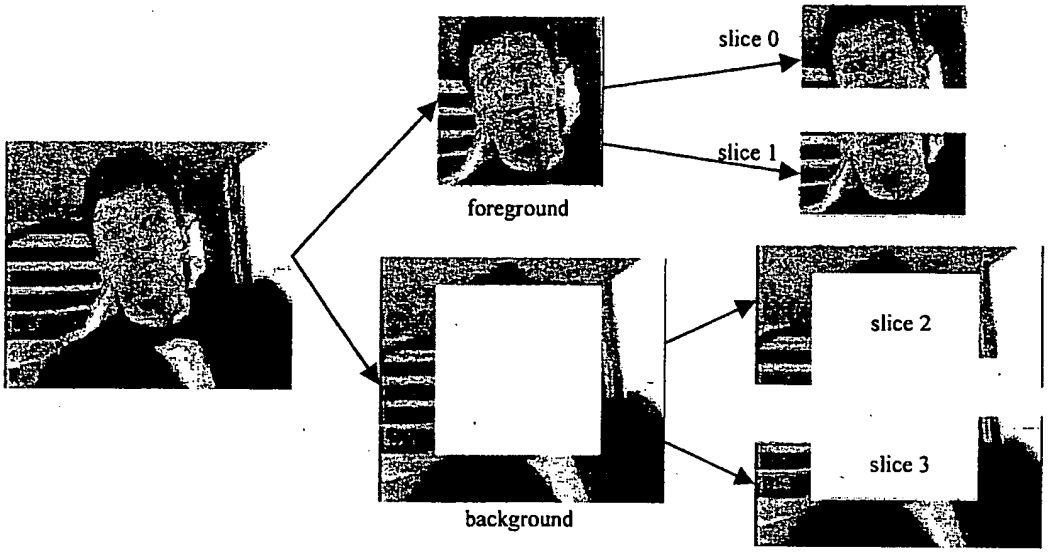


Figure 1. Example of picture, sub-picture, and slice structures.

treat all regions equally. Arbitrarily shaped objects [9], as defined in the MPEG-4 standard, can be used to extract the regions of interest. The object shape is represented by a binary alpha map and the object transparency can be represented by an alpha plane. Objects are then laid on top of each other to compose a picture to display.

1.3. Overview of the Paper

While MPEG-4 arbitrary shape coding has proven to be efficient in compression terms for many sequences, it often inherits relatively high processing requirements. This paper proposes a simple region-based coding method, called sub-picture coding, which is based on rectangular shapes. It can be considered as a simplification of arbitrary shape coding. While sub-picture coding does not provide means to define object boundaries accurately, it suits unequal error protection.

Section 2 of the paper presents the sub-picture coding method. The operation of the sub-picture coding method was simulated in multicast Internet streaming conditions, as explained in Section 3. The paper is concluded in Section 4.

2. SUB-PICTURE CODING

The proposal adds a sub-picture coding layer between picture and slice layers. Sub-pictures are rectangular except for the so-called background sub-picture, which consists of the picture area not falling to any of the rectangular sub-pictures. Rectangular sub-pictures are also referred to as foreground sub-pictures. Sub-pictures boundaries are aligned with macroblock boundaries, and sub-pictures do not overlap. A slice resides within one sub-picture only. A slice in a background sub-picture may not contain spatially adjacent macroblocks within a macroblock line, as it can be intervened by foreground sub-pictures. Figure 1 shows an example of a picture having one foreground sub-picture and one background sub-picture, both of which include two slices.

There are two coding modes associated with sub-pictures: normal prediction mode and independent sub-picture decoding mode. In the normal prediction mode, sub-picture boundaries are treated as slice boundaries. In the independent sub-picture decoding mode, boundaries of foreground sub-pictures are treated as image boundaries and sub-picture segmentation is static over a group of pictures similarly to H.263 Independent Segment Decoding. In other words, temporal and spatial prediction over sub-picture boundaries is prevented when coding rectangular sub-pictures. No such limitation exists when coding a background sub-picture, as the background sub-picture is considered to have a lower subjective importance and it is not protected against error propagation.

3. SIMULATIONS

3.1. Conditions

We selected multicast Internet streaming as a target application. As interactive error concealment cannot be used in large scale with IP multicast, forward error control methods have to be used. Thus, simulation conditions can

BEST AVAILABLE COPY

TML with sub-picture coding TML with ROI quantization TML with constant quantization

Figure 2. Example snapshots, 20 % packet loss rate.

be close to simulation conditions for IP video conferencing without interactive error control methods. We used the ITU-T Video Coding Experts Group (VCEG) common conditions for the low-delay Internet applications [10] as much as we considered appropriate. In addition, we applied transport coding level forward error correction according to RFC 2733 [3].

ITU-T VCEG has been working on the H.26L video codec project. A Joint Video Team (JVT) was recently established joining the experts from ITU-T VCEG and ISO/IEC Moving Picture Experts Group (MPEG). JVT develops a codec based on the draft H.26L design.

The sub-picture coding scheme was implemented in and tested with H.26L Test Model Long-Term (TML) version 8.6 [11][12]. A constant foreground sub-picture was selected, and a finer quantizer was used in the foreground region. The independent sub-picture decoding mode was in use. The scheme was compared to TML-8.6 with region-of-interest (ROI) quantization and to TML-8.6 with constant quantization. The quantization parameters for the ROI coding scheme were obtained in the same manner as for the sub-picture coding scheme but no sub-pictures were used.

The core of the simulator consisted of the following phases. First, the TML encoder was used to encode a sequence. The encoder generated an RTP packet stream. Then, a packet loss simulator erased some of the generated packets according to error patterns released in [13]. The resulting stream of packets was decapsulated and decoded using the TML decoder. Finally, the peak signal-to-noise ratio (PSNR) was calculated between each frame of the source sequence (at full frame rate) and the corresponding reconstructed frame.

Encapsulation into RTP packets was done as follows: In the sub-picture coding scheme, INTRA pictures were encapsulated into five packets. There were two packets for the foreground sub-picture: one packet contained odd macroblock rows and another packet contained even macroblock rows. This slice interleaving mechanism, introduced in [14], was used to obtain a better error concealment result. One parity FEC packet was generated for the two foreground packets according to RFC 2733. The background sub-picture was packetized into another two packets using slice interleaving method. Two consecutive INTER pictures consisted a group, and for each such group there were two foreground sub-picture packets, one parity FEC packet for the foreground packets, and two background sub-picture packets. A sub-picture packet contained data from two pictures: macroblocks from even rows of a certain frame and macroblocks from odd rows of the next frame or vice versa. When sub-picture coding was not in use, there were three packets for each INTRA and INTER frame: two packets for the entire picture (slice interleaving applied), and one parity FEC packet for the two packets.

As multicast Internet streaming was assumed, the coding and transport scheme was tailored for the worst expected case. We selected optimal coding parameters and encapsulation options by trial and error according to the PSNR performance in the 20 % packet loss rate. We tested several INTRA macroblock coding rates as well as several slice sizes to find the optimal ones. The packet stream obtaining the best PSNR performance was multicast virtually. The PSNR performance in 0, 3, 5, and 10 % packet loss rate was then simulated with the multicast packet stream. In order to obtain statistically significant results, each packet stream was virtually transported ten times for each packet loss rate. The first loss pattern position of a run starts from where the previous run ended.

3.2. Results

The experiments were done with the Carphone, Hall Monitor, and Coastguard sequences. We present the results for the Carphone sequence only due to lack of space. The results with the other sequences were consistent with the Carphone results.

Figure 2 shows an example snapshot of the Carphone sequence in 20 % packet loss rate. It can be seen that the sub-picture coding scheme maintains the best image quality. A clearly visible error hit the stream coded with the ROI quantization scheme. When constant quantization

BEST AVAILABLE COPY

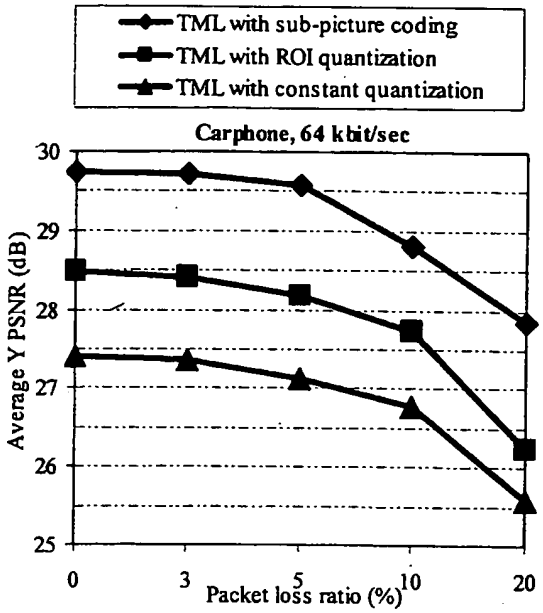


Figure 3. Average luminance PSNR of foreground.

was used, the foreground area was quantized more coarsely than in the other cases, which can be easily seen from Figure 2.

Figure 3 presents the average luminance PSNR of the foreground area of the Carphone sequence. These objective results confirm the subjective ones. Sub-picture coding improves the average luminance PSNR of the foreground area more than one decibel regardless of the packet loss rate. This improvement is gained at the expense of background image quality, which is degraded by coarser quantization and less error protection. In fact, the overall PSNR in the sub-picture coding case drops a little compared to the ROI quantization case and somewhat more compared to the constant quantization case. However, errors in the background are far less noticeable than errors in the foreground, and therefore the overall subjective quality is improved.

4. CONCLUSION

Many video communication systems can make use of unequal error protection to improve quality while maintaining or decreasing bit-rates. In order to apply unequal error protection, video bit-streams have to be organized in portions of different importance in terms of visual quality. Techniques achieving this goal include data partitioning scalable video coding, and region-based video coding. This paper introduced a simple region-based video coding method called sub-picture coding. It provides means for unequal error protection based on regions of interest. In sub-picture coding, a picture is partitioned to one or more rectangular foreground sub-pictures and to a hollow background sub-picture along macroblock boundaries, and sub-picture edges are treated as slice boundaries or as picture boundaries depending on the signaled coding mode. The sub-picture coding method was simulated in multicast Internet streaming conditions and it was found to be superior to conventional coding. Due to its simplicity and the obtained results, we think that sub-picture coding has some potential to become one of the key video coding tools for unequal error protection.

5. REFERENCES

[1] A. Miyazaki, H. Fukushima, K. Hata, T. Wiebke, R. Hakenberg, C. Burmeister, N. Takatori, S. Okumura, and T. Ohno, "RTP Retransmission Payload Format", IETF Internet Draft draft-ietf-avt-rtp-selret-02.txt, 18 July 2001.

[2] D. Leon, V. Varsa, "RTP retransmission framework", IETF Internet Draft draft-leon-rtp-retransmission-00.txt, July 2001.

[3] J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction" IETF RFC 2733, December 1999.

[4] 3GPP TS 23.107, "QoS Concept and Architecture (Release 4)", V4.2.0, October 2001, available via ftp://ftp.3gpp.org/Specs/2001-09/Rel-4/23_series/23107-420.zip.

[5] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", IETF RFC 2474, December 1998.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services", IETF RFC 2475, December 1998.

[7] M. M. Hannuksela, "Simple Packet Loss Recovery Method for Video Streaming", International Packet Video Workshop PV2001, Kyongju, South Korea, 30 April – 1 May 2001.

[8] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 3, March 2001.

[9] N. Brady, "MPEG-4 Standardized Methods for the Compression of Arbitrarily Shaped Video Objects", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 8, December 1999.

[10] S. Wenger, "Common conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing", ITU-T Video Coding Experts Group, document VCEG-N79r1, September 2001.

[11] T. Wiegand (editor), "H.26L Test Model Long-Term Number 9 (TML-9) draft0", ITU-T Video Coding Experts Group, document VCEG-N83d1, December 2001.

[12] H.26L Test Model Software, version 8.6, October 2001.

[13] S. Wenger, "Error patterns for Internet video experiments", ITU-T SG16 document Q15-I-16-R1, October 1999.

[14] S. Wenger and G. Côté, "Using RFC2429 and H.263+ at low to medium bit-rates for low-latency applications", International Packet Video Workshop PV99, April 1999.

ITU - Telecommunications Standardization Sector
STUDY GROUP 16 Question 6
Video Coding Experts Group (VCEG)

Document VCEG-O57
Filename: VCEG-O57.doc
Generated: 16 Jan '02

15th Meeting: Pattaya, Thailand, 3-7 December, 2001

Question: Q.6/SG16 (VCEG)

Source:

Ye-Kui Wang
TICSP
Tampere Univ. of Technology
P.O. Box 553
FIN-33101 Tampere
Finland
Tel: +358 3 3115 3972
Email: wyk@ieee.org

Miska M. Hannuksela,
Nokia Mobile Phones
P.O. Box 88
33721 Tampere
Finland
Tel: +358 40 521 2845
Email: miska.hannuksela@nokia.com

Title: Core Experiment Description of Sub-Picture Coding

Purpose: Information

---

1. INTRODUCTION

In the Pattaya meeting, VCEG/JVT decided to initiate a core experiment about the sub-picture coding technique presented in VCEG-O46, titled "New Image Segmentation Method". This document describes the core experiment. It also includes an executable simulation script, the source code, and some explanations to ease the experiment. The script and the source code are contained in the accompanying files "VCEG-O57_script.zip" and "VCEG-O57_src.zip", respectively.

Section 2 of the document reprints the description of the sub-picture coding method for convenience. The coding method is unchanged compared to VCEG-O46. The core experiment is targeted for Internet multicast. Section 3 describes the simulation conditions, which are essentially the same as in VCEG-O46 except for improved quantization parameter allocation. Section 4 provides instructions how to run the simulation script and some remarks on the source code.

2. SUB-PICTURE CODING

2.1 Sub-picture Layer

A sub-picture coding layer is added between picture and slice layers. Sub-pictures are rectangular except for the so-called background sub-picture, which consists of the picture area not falling to any of the rectangular sub-pictures. Rectangular sub-pictures are also referred to as foreground sub-pictures. Sub-pictures do not overlap. All sub-pictures are coded in scan-order. A slice resides within one sub-picture only. A slice in a background sub-picture may not contain spatially adjacent macroblocks within a macroblock line, as it can be intervened by foreground sub-pictures.

2.2 Coding Modes

There are two coding modes associated with sub-pictures: independent sub-picture coding and normal prediction mode.

In independent sub-picture coding, boundaries of rectangular segments are treated as image boundaries. Sub-picture segmentation is static over a group of pictures (or any similar grouping of pictures). Temporal and spatial prediction over sub-picture boundaries is prevented when coding rectangular sub-pictures. No such limitation exists when coding the background sub-picture, as the background sub-picture is considered to have a lower subjective importance and it is not protected against error propagation. Motion vectors shall not point outside rectangular sub-pictures. Spatial prediction or loop-filtering shall not be done across sub-picture boundaries.

In normal prediction mode, sub-picture boundaries are treated as slice boundaries. Motion vectors may point outside sub-pictures freely and loop-filtering is applied across sub-picture boundaries.

2.3 Bit-Stream Syntax

Slices (or data partitions) are considered the minimum separable unit for transport packetization. We did not want to introduce a competitive concept to slices in the bit-stream level, and therefore we included the sub-picture information in the slice header. Consequently, the transport packetization and depacketization process work the same way as previously regardless of whether sub-pictures are in use.

Use of sub-pictures is signaled in the parameter list of picture and sequence layer data (see VCEG-N72-R1).

When sub-pictures are in use, the slice header shall be as follows:

PictureID	As defined in VCEG-N72-R1.
SliceType	As defined in VCEG-N72-R1.
FirstMBInSliceX	The horizontal position (column) of the first macroblock in the slice relative to the sub-picture.
FirstMBInSliceY	The vertical position (row) of the first macroblock in the slice relative to the sub-picture.
InitialQP	As defined in VCEG-N72-R1.
SubPictureID	Unique identifier of the sub-picture. Each sub-picture is assigned an ID starting from zero and incremented by one in coding order. The count shall be reset for each picture. If independent sub-picture coding is in use, sub-picture ID shall remain the same for the spatially matching sub-pictures over a group of pictures.
SubPictureInfo	0: Sub-picture attributes are the same as the attributes of a sub-picture having the same ID in the previous picture. This value is useful especially in the independent sub-picture coding mode. 1: Sub-picture attributes are the same as the attributes of a sub-picture having the same ID in the same picture. This value is used if a sub-picture contains multiple slices. 2: Sub-picture location and size is defined in the following four codewords. If independent sub-picture coding is in use, the following four codewords shall remain the same within a group

of blocks. A repetition of the codewords is allowed for error resiliency purposes.
3: Background sub-picture. If one of the earlier sub-pictures for the same picture is lost and its location and size are not externally signaled (which is typical in normal prediction mode), decoder shall not decode the background sub-picture, as its shape is unknown.

Left — The coordinate of the left-most macroblock in the sub-picture (in macroblocks). The left-most macroblock column of the picture is assigned value zero.

Top — The coordinate of the top-most macroblock in the sub-picture (in macroblocks). The top-most macroblock row of the picture is assigned value zero.

Width — Width of the sub-picture. The codewords are assigned as follows:

Symbol no	UVLC code	Explanation
0	1	Guess = (RightMost - Left) / 2 + 1, where RightMost is the column address of the right-most macroblock of the picture and / stands for division by truncation. For example, for a QCIF picture and Left equal to 3, Width becomes (10-3)/2+1 = 4.
1	001	Guess + 1
2	011	Guess – 1
3	00001	Guess + 2
4	00011	Guess – 2
...	...	...

Height — Height of the sub-picture. The codewords are assigned similarly to Width.

3. SIMULATION CONDITIONS

3.1 Overview

The performances of three codecs are to be compared:

1) TML-8.6 with the sub-picture coding scheme proposed in VCEG-O46 (herein referred to as the RSP codec),

2) TML-8.6 with region-of-interest (ROI) quantization (ConvROI codec), and

3) the original TML-8.6 without ROI quantization (Conv codec).

Conditions simulating multicast Internet streaming are going to be applied. The conditions are based on the common conditions for the low-delay Internet applications (VCEG-N79R1). As interactive error concealment cannot be used in large-scale with IP multicast, parity FEC according to RFC 2733 is used for each codec.

The simulation procedure for each input sequence consists of two main phases:

1. As a single packet stream is transmitted in multicast Internet streaming, coding and transport parameters are optimized by trial and error for the worst expected case (20 %

packet loss rate). Several INTRA GOB update (IGU) rates as well as several slice sizes are tested. The packet stream obtaining the best PSNR performance is selected. Section 3.3 describes which algorithm is used in IGU rate and slice size optimization. Section 3.4 explains how the bit-rate is controlled and how quantization parameters are assigned.

2. The packet stream is multicast virtually, and the PSNR performance in 0, 3, 5, and 10 % packet loss rate is simulated.

Both main simulation phases include packetization, packet loss simulation according to VCEG-N79r1, and depacketization. Section 3.5 gives an insight into the packetization process.

Instead of encoding 4000 frames as specified in VCEG-N79R1, the PSNR of the decoded video is calculated for each of the 10 runs, the average PSNR plus the best and worst cases of the 10 runs should be shown, as proposed in VCEG-M77. This method is used to show the variation of the PSNR depending on the position of the loss pattern files. The first loss pattern position of a run starts from where the previous run ended. PSNR is calculated between each frame of the source sequence (at full frame rate) and the corresponding reconstructed frame, including repetitions of previous frames in place of skipped and lost frames.

Earlier, for VCEG-O46, we carried out simulations without transport-layer FEC coding too (results for Carphone included in VCEG-O46 and results for Coastguard and Hall presented in the Pattaya meeting). However, when FEC was applied, the results were considerably better than those were without FEC. Moreover, a system providing transport-layer unequal error protection is better to demonstrate the potential of the sub-picture coding method. Therefore, the core experiment is carried out with a system applying FEC according to RFC 2733.

3.2 Test Sequences

The sequences, picture sizes, frame rates, bit-rates, and segmentation of the foreground sub-pictures, described in this section are recommended in the core experiment.

Five QCIF sequences (Carphone, Coastguard, Foreman, Hall and News) and one CIF sequence (Irene), with different frame rates and bitrates, are simulated. Foreman sequence is simulated twice with different bitrates. Therefore, there are of totally seven simulation cases, as shown in Table 1. The foreground sub-picture position and size for each case are also shown in Table 1.

Table 1. The seven simulation cases.

Sequence	Bitrate (kbps)	Frame Rate (frame/s)	Num of Encoded Frames	Foreground Sub-picture Position & Size (Unit: MB)			
				Left	Top	Width	Height
Carphone	64	10	99	2	1	6	6
Coastguard	64	10	99	1	2	9	4
Foreman	64	7.5	75	2	2	7	6
Foreman	144	7.5	75	2	2	7	6
Hall	32	10	99	2	2	7	4
News	64	15	149	2	1	8	6
Irene	384	30	99	4	2	11	13

The foreground sub-picture for each sequence is obtained by manual segmentation, which tries to find the best common region of interest for all used frames in the sequence. The visual segmenting results of the six sequences can be roughly seen from Figure 1.

Figure 1. Sub-picture segmentation for the test sequences.

3.3 Optimization of Slice Size and IGU rate

For QCIF sequences, the slice size is selected from the set of 1, 3, or 9 GOBs/slice (the last means 1 slice/picture), and the IGU rate is 1 GOB per 1, 2, 3, 4, 5, or 0 pictures (the last means that no GOBs are forced to be intra-coded). For CIF sequences, the slice size is selected from the set of 1, 3, 9, or 18 GOBs/slice (the last means 1 slice/picture), and the range of possible IGU rates is the same as for QCIF sequences. It is commonly known that both smaller slice size and higher IGU rate typically result into better error resilience.

Since exhaustive optimization methods are too time-consuming, the following optimization strategy is applied:

- Foreground parameters are optimized first. Then, the background parameters are optimized while the foreground parameters are fixed to the ones found in the first phase. The underlying reason for this strategy is that background slices can be predicted from foreground slices, while foreground slices cannot be predicted from background slices. Therefore, background parameters cannot affect foreground coding very much. This rule only applies to the RSP codec.

- IGU rate is optimized before slice size. We think that IGU rate imposes a heavier effect on error resilience than slice size. When optimising IGU rate, the most error resilient slice size, i.e. 1 GOB/slice, is used.

- When optimizing a parameter (slice size or IGU rate), the parameter values are tested in descending order of error robustness, i.e., from 1 GOBs/slice to 9 GOBs/slice and from 1 forced intra GOB/picture to no forced intra GOBs. Each time the PSNR performance degrades compared to the previous parameter value, optimization of the parameter is stopped without testing the remaining parameter values.

PSNR values of both foreground sub-pictures and entire pictures are calculated for each codec. Since foreground is to be better protected, selection of better parameters cannot be PSNR of the entire pictures. Meanwhile, background should also be considered. Therefore, the following weighted PSNR is used in comparing different results:

$$wPSNR = (PSNRf * 3 + PSNRb) / 4$$

BEST AVAILABLE COPY

The parameter combination with the highest wPSNR is selected. Only the luminance component is considered.

3.4 Selection of Quantization Parameter

In VCEG-O46, adjusting QPs of the foreground and background for the RSP codec and the ConvROI codec focused only on good foreground PSNR performance. As a result, in the two sides of the foreground boundary, the QP difference was likely to be large, which caused annoying boundary effects in the decoded pictures. To reduce the boundary effects, a new scheme described in the following is applied.

The background QP, QPb, is fixed for all background MBs. The foreground QP is a function of the distance from edge between the foreground sub-picture and the background sub-picture. For QCIF sequences, if a macroblock is adjacent to the edge, its QP is QPb-1. If the distance between a macroblock and the edge is one macroblock, the QP of the macroblock is QPb-2. For other foreground macroblocks, the QP is QPb-3. An example is shown in Figure 2, where the highlighted MBs form the foreground sub-picture.

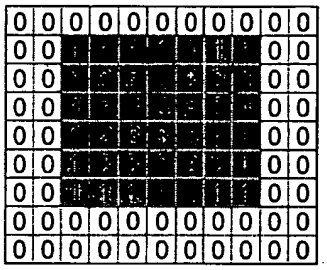


Figure 2. Example of the quantization scheme for QCIF sequences.

The scheme for CIF sequences is similar. If the distance between a macroblock and the foreground-background edge is zero or one, the QP of the macroblock is QPb-1. If the distance between a macroblock and the foreground-background edge is two or three, the QP of the macroblock is QPb-2. For other foreground macroblocks, the QP is QPb-3. An example is shown in Figure 3, where the highlighted MBs form the foreground sub-picture.

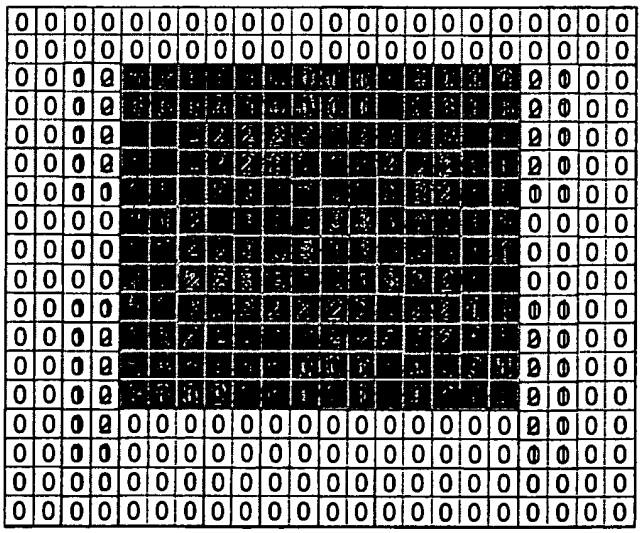


Figure 3. Example of the quantization scheme for CIF sequences.

BEST AVAILABLE COPY

Quantization parameters for the ConvROI codec are obtained in the same manner as for the sub-picture coding scheme.

3.5 Packetization

The packetization method is slightly different from what is described in VCEG-O46, as follows:

The RSP codec:

- For the intra picture, there are 5 packets: 2 packets for the foreground sub-picture, 1 parity FEC packet for the foreground packets, and 2 packet for the background sub-picture. The slice interleaving mechanism (one packet contains odd macroblock rows and another packet contains even macroblock rows) is applied in both the two kinds of sub-pictures.

- For inter pictures, each successive 2 frames consist a group. For each group, there are 2 foreground sub-picture packets, 1 parity FEC packet for the foreground packets, and 2 background sub-picture packets. Each sub-picture packet contains data from two pictures: macroblocks from even rows of frame n and macroblocks from odd rows of frame n+1 or vice versa.

The conventional codecs (the ConvROI codec and the Conv. codec):

- There are 3 packets for each intra and inter frame: 2 packets for the entire picture (slice interleaving applied), and 1 parity FEC packet.

4. PROVIDED SIMULATION SOFTWARE

4.1 Step-By-Step Guide to Run the Simulation

The executable script has been tested with Windows 2000 and NT4.

Having the file "VCEG-O57_script.zip", please do as the following steps:

1) Decompress the contained files to the directory that you want to be the working directory for the experiment.

2) Create/modify the input parameter files (e.g. "sim_foreman64.dat") for the sequences you are going to simulate. The explanation of each input parameter is provided in the example parameter files. You should prepare for each simulating case (a sequence with other parameters, e.g. bitrate, fixed) an input parameter file.

3) Modify the batch file "RSPvidcodsim.bat", to add all planned simulating cases.

4) Run the batch file to begin the simulation.

5) After the simulation process is finished, simulation results of all the simulating cases are saved in the sub-directories of the working directory:

   - Bitrates information (desired bitrate, available video bitrate, and the resulting video bitrate), QP, and optimized slicing method(s), IGU rate(s) are provided in the sub-directory of "parameters".

   - Objective results (PSNR values of foreground sub-pictures and entire pictures) are saved in the sub- directory "PSNR".

- The decoded sequences can be found in the sub- directory of "decSeq".
- Other sub-directories are for saving the error pattern files ("errorpattern"), the encoded bitstreams ("bitstream"), and the temporary foreground sub-picture sequences ("subSeq").

Note that enough free disk space should be provided for the experiment, mainly to save the decoded sequences. The amount of required disk space depends on how many simulating cases and how many frames to be encoded in each case.

4.2 Remarks on Source Code

4.2.1 Saving the Decoded Sequence

To compare the subjective results, decoded sequences should be saved. However, each encoded bitstream file with optimized parameters will be decoded 10 times, by applying 10 continuous runs of the used error pattern file. The decoded sequence of the run whose wPSNR is closest to the averaged wPSNR is saved.

In each decoded sequence, the number of frames is exactly the same as the number of encoded frames. That is, the corresponding reconstructed frames (repeat of the previous reconstructed frames) of lost frames are also saved. This is beneficial for comparing decoded sequences under different packet loss rate when evaluating the subjective results.

4.2.2 Considering Bits of FEC Packets in Resulting Bitrate

When FEC is used to protect the bitstream in transmission, the bits of FEC packets should be considered as part of the resulting video bitrate. As we know, the size of the parity FEC packet equals to the larger size of the two packets to be protected. This is implemented in the encoder function encode_one_frame(), after the current frame is encoded.

4.2.3 Lossy Channel Simulation

The method to apply the packet loss patterns to the bitstream files is really a "simulation". The packet loss simulator is implemented in the decoder software project, by the function Simulator_H26L_Slice() and some functions it calls. No packet is really formed. The simulator identifies to which packet each slice belongs according to the packetization method, and identifies whether a packet is lost according to the used packet loss pattern. The slices belong to the lost packets are discarded, and the slices belong to the received packets are written into the resulting bitstream file, which will be taken as input by the decoder.

4.2.4 Decoding and Sub-Picture Error Concealment Strategy in the RSP Codec

The order of the sub-pictures in the encoded bitsream is: SP0, B0, SP1, B1, SP2, B2, SP3, B3, SP4, B4, ... , where SPn and Bn means the foreground and background sub-picture, respectively, of frame n. The sub-picture error concealment strategy, therefore the framework of the decoder, is designed according to the current decoded sub-picture and the sub-picture losses, as described in the following:

- If current decoded sub-picture is a foreground sub-picture (SPn), the next sub-picture might be:

1) Bn
   No sub-picture is lost. The decoding steps are:
   a. Decode SPn, do error concealment if slice loss exists.
   b. Mark the status of the foreground sub-picture for the next received background sub-picture as "received". Note that the status of any foreground sub-picture SPx affects the error concealment process of its background sub-picture Bx.
2) SP(n+m), m is larger than 0 (1, 2, 3, ...)
   At least Bn is lost, and entire frames might be lost. The decoding steps are:
   a. Decode SPn, do error concealment if slice loss exists.
   b. For frame n, copy SPn to the foreground region of Bn, then mark the MBs of the background region as corrupted and do error concealment. Note that here the MBs of the foreground should be marked as received. The simplest method to form the decoded picture of frame n is to directly copy the background region from B(n-1), plus the foreground region from SPn. Since no bitstream exist for Bn, a virtual frame decoding function is called. This is implemented as the decoder function decode_one_frame_virtual().
   c. Adjust the reference buffers according to the number of entirely lost frames. This is implemented in the decoder function init_frame(), which will be called by the next sub-picture decoding process.
3) B(n+m), m is larger than 0 (1, 2, 3, ...)
   At least Bn and SP(n+1) are lost, and entire frames might be lost. The decoding steps are:
   a. All the three steps in the previous case.
   b. Mark the status of the foreground sub-picture for the next received background sub-picture as "lost". Therefore in the end the next decoding process for B(n+m), the foreground region could be error concealed.
4) End of sequence (EOS)
   Similar to case 2), carry out its first two steps.

- If current decoded sub-picture is a background sub-picture (Bn), the next subpicture might be:

1) SP(n+1)
   No sub-picture is lost. Therefore, the only decoding step is to decode Bn, do error concealment if slice loss exists.
2) B(n+m), m is larger than 0 (1, 2, 3, ...)
   At least SP(n+1) is lost, and entire frames might be lost. The decoding steps are:
   a. Decode Bn, do do error concealment if slice loss exists.
   b. Adjust the reference buffers according to the number of entirely lost frames.
   c. Mark the status of the foreground sub-picture for the next received background sub-picture as "lost".
3) SP(n+m), m is larger than 1 (2, 3, 4, ...)
   There are m-1 frames entirely lost. The decoding steps are the two steps in the previous case.
4) End of sequence (EOS)
   Entire frames might be lost. Nothing can be done more than to decode Bn and do error concealment if slice loss exists.

4.2.5 PSNR Calculation

According to VCEG-N79, PSNR should be calculated between each and every frame of the source sequence (at full frame rate) and the corresponding reconstructed frame, including repetitions of previous frames in place of skipped and lost frames. PSNR calculation for the skipped frames are implemented in the decoder function findSNR4skippedFrames() and findSNR4skippedFrames_s(), the later is for the foreground sub-picture. PSNR calculation for the lost frames are implemented in the decoder function init_frame().

4.2.6 Hacks

There are four hacks in the source code:

1) *Bitstream Syntax Implementation.* Since TML-8.6 does not follow VCEG-N72r1, the slice header syntax in the source code is not the same as described in subsection 2.3. Instead, the sub-picture position and size information are implemented in the picture header. This should not affect the results significantly.

2) *No packet Loss in the Intra frame.* Since only the first frame is intra coded for each encoded sequence, it is possible that one decoding case has lost a packet in the intra frame, while the others do not have. We feel that this might affect the objective and subjective results in an unfair way, as ten runs might not be enough to smooth out the quality drop. Therefore, in the decoded sequences, all the packets of the first frame (the Intra frame) are assumed to be correctly received. This is realized by the packet loss simulator. Each time when there is at least one packet loss in the Intra frame, the simulator will discard the obtained error patterns and read another set of error patterns from the used error pattern file.

3) *P_interval for PSNR calculation.* To calculate PSNR for the skipped frames, the number of skipped frames (therefore the variable P_interval in the source code of TML decoder) should be known at first. In the TML decoder, P_interval is calculated by subtracting the temporal reference (TR) of one P-frame/I-frame from the next P-frame. Therefore, it can be known only after the first P-frame (the second encoded frame) is received. In other words, P_interval cannot be known just after decoding of the first frame. As the result, PSNR of the skipped frames following the first frame cannot be calculated. In the provided software, P_interval is calculated by reading the skipped frames number from the input parameter file (e.g. "sim_foreman64.dat").

4) *Hack for "assert (coef_ctr < 4)".* There is a very "old" bug in the TML decoder, initiated from TML-8.0. That is, in the decoder function readCBPandCoeffsFromNAL(), which can be captured by the instruction line "assert (coef_ctr < 4);" in the function body. The bug occurs in very low frequency (therefore is difficult to debug and to fix). However, each times it occurs, it will cause "unfortunate" result and halt the program! Since the authors still have not found a fix, they implemented a hack: when the bug occurs, the decoding process is stopped, and the bitstream is discarded by transferring to the main simulating program an average PSNR value of 0 dB. We are hoping someone can fix this bug as soon as possible!

НОУ № 11750
V

Б.А. Локшин

ЦИФРОВОЕ ВЕЩАНИЕ:

ОТ СТУДИИ К ТЕЛЕЗРИТЕЛЮ

(B)

Под редакцией проф. Л.С. Виленчика

Инв. № 55555

Москва 2001

BEST AVAILABLE COPY

ны в левом верхнем углу матрицы. Наиболее эффективный метод считывания значений коэффициентов — зигзагообразное сканирование из верхнего левого в правый нижний угол (рис. 2.7), при котором сначала прочитываются все ненулевые, а затем нулевые коэффициенты. Для кодирования последних применяется описанный выше метод кодирования длин серий, при котором каждой непрерывной последовательности одинаковых символов (в данном случае нулей) приписываются отдельные кодовые слова.

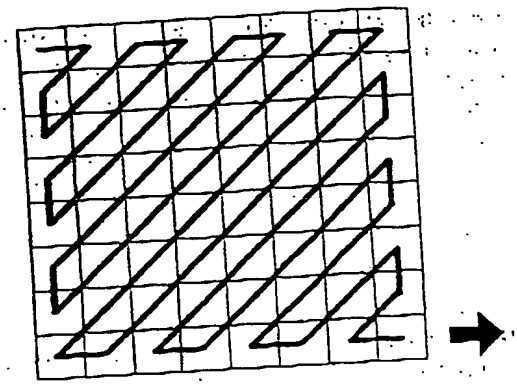


Рис. 2.7. Зигзагообразное сканирование коэффициентов ДКП

Статистическая вероятность появления различных значений коэффициентов неодинакова: некоторые появляются очень часто, другие редко. Эта особенность может быть использована для дальнейшего сокращения числа битов, если более часто встречающимся значениям присвоить более короткие последовательности символов, а более редким — более длинные. Эту задачу решают коды с переменной длиной, о которых мы упоминали выше.

В зависимости от сюжета изображения и скорости движения объектов доля устраняемой избыточности изменяется и среднее число битов, приходящееся на один отсчет, может варьироваться в широких пределах. В таких же пределах будет изменяться и скорость цифрового потока на выходе кодера сжатия. В то же время большинство каналов связи рассчитаны на передачу сигналов с постоянными скоростями. Как всегда в таких случаях, выручает буферная память. Буфер накапливает избыточные биты в критические для кодера моменты (смена сюжета, быстрое движение) и отдает их в канал в периоды относительного спокойствия, когда канал оказывается недогруженным текущими данными. Предусмотрен и механизм обратной связи на случай переполнения или, наоборот, опустошения буфера: изменяя адаптивно коэффициенты матрицы квантования, буфер может в определенных пределах управлять количеством поступающих битов и скоростью потока, разумеется, ценой соответствующего увеличения или уменьшения шага квантования и связанных с ним шумов квантования.

2.1.5. Некоторые действующие стандарты цифрового сжатия

Разработанные в конце 80-х годов эффективные методы цифрового сжатия изображений, описанные выше, вызвали к жизни большое число схем обработки, сочетающих различные методы и направленных на решение разных задач. Часть этих схем была принята стандартизирующими органами и получила статус международных стандартов. Рассмотрим наиболее эффективные из них, послужившие базой для стандартов семейства MPEG.

В 1982–1990 годах группой экспертов по кодированию неподвижных изображений (Joint Photographic Expert Group), образованной совместно МОС, МЭК и МСЭ, был разработан и затем принят одноименный стандарт кодирования неподвижных изображений [2.6]. Основной задачей было поставлено сокращение необходимого объема данных для передачи высококачественного изображения (до 0,25 бит/пикс. для «различимого» изображения, 1 бит/пикс. для «отличного» качества и 4 бит/пикс. для «неотличимого» от оригинала, при исходной точности квантования 8 бит/пикс.).

В ходе длительных исследований были опробованы многие алгоритмы цифрового

BEST AVAILABLE COPY

сжатия и выбрано ДКП как обеспечивающее наилучшее качество при минимальной скорости цифрового потока. В стандарте JPEG впервые были введены технические решения, вошедшие затем в последующие стандарты: обработка не исходных цветовых, а матрицированных сигналов яркости и цветоразностных сигналов, прореживание последних вдвое до формата «4:2:0», выбор размера блоков ДКП 8 x 8 пикс, зигзагообразное сканирование коэффициентов, раздельное квантование постоянной составляющей и остальных коэффициентов ДКП, энтропийное кодирование. По понятным причинам отсутствовала только компенсация движения.

Следующий представляющий интерес стандарт под названием H.261 был разработан в 1988–90 гг. предшественником нынешнего МСЭ-Т Международным консультативным комитетом по телеграфии и телефонии. Стандарт предназначался для передачи видеоизображений (видеотелефония, видеоконференции) со скоростями n x 64 кбит/с, где n = 1...30, формат изображения не мог превышать CIF. В этом стандарте уже использовалось кодирование с предсказанием и два типа кадров: I- и P-кадры. Предсказание движения осуществлялось сравнением с предшествующим кадром, в качестве критерия сопряжения был принят минимум абсолютной ошибки. Начальное окно поиска составляло прямоугольный участок размером 64 x 48 пикс., поиск мог осуществляться одним из трех способов: 1) полным перебором всей области поиска (медленный способ); 2) путем последовательного сужения окна, пока не достигалась требуемая точность сопряжения; 3) иерархической оценкой движения, при которой формируются несколько последовательных версий опорного и текущего кадров со все более низким разрешением, ищется наилучшее согласование в версии с самым низким разрешением и пересчитывается к исходному уровню путем последовательной модификации векторов перемещения. Как и в современных стандартах, макроблок размером 16 x 16 пикс. содержал 4 блока отсчетов сигнала яркости и по одному блоку цветоразностных сигналов.

Стандарты H.261 и JPEG легли в основу разработки MPEG-1.

Работа в этом направлении была продолжена, и в 1996 г. МСЭ-Т принял для тех же применений стандарт H.263, более приспособленный для низкоскоростного кодирования. В этом стандарте поддерживаются форматы от SQCIF до 16CIF (1408 x 1152 пикс.), присутствуют все три вида кадров I, P и B, область поиска сопряжения не ограничена, точность сопряжения 0,5 пикс. К основным улучшениям по сравнению с H.261 относятся: неограниченный диапазон изменения векторов перемещения (на границе они могут

Табл. 2.1. Некоторые форматы разложения, поддерживаемые стандартами H.261 и H.263

Формат	Число отсчетов сигнала яркости	Число отсчетов сигнала цветности	Стандарт H.261	Стандарт H.263
SQCIF	128x96	64x48	по заказу	поддерживается
QCIF	176x144	88x72	поддерживается	поддерживается
CIF	352x288	176x144	по заказу	по заказу
4CIF	704x576	352x288	не поддерживается	по заказу
16CIF	1408x1152	704x576	не поддерживается	по заказу

Page 907 of 2504 BEST AVAILABLE COPY

указывать за пределы области изображения), арифметическое кодирование квантованных коэффициентов ДКП, улучшенное предсказание (предсказываются 4 блока векторов 8 x 8 вместо одного размерностью 16 x 16 пикселов), кодирование PB-кадров (последовательные P- и B-кадр могут кодироваться вместе как один PB-кадр). Кодек H.263 вышел за границы скорости 64 кбит/с и стал кодеком широкого применения для низкоскоростного кодирования

В табл. 2.1 приведены некоторые параметры стандартов H.261 и H.263 для различных форматов изображения.

Появившаяся недавно вторая версия стандарта H.263, названная H.263+, содержит много дополнительных функций, делающих этот стандарт вполне современным и эффективным: усовершенствован режим внутрикадровой компрессии (деблокирование фильтра); предсказание делается на основе передискретизированного опорного изображения; введен режим обновления с пониженным разрешением; использованы реверсивные коды переменной длины; формат изображения и тактовые частоты могут устанавливаться по выбору пользователя; введен режим временной, пространственной масштабируемости и масштабируемости отношения сигнал-шум; повышена устойчивость к ошибкам благодаря структурированию слайсов.

Отметим, что стандарт H.263+ использовался при разработке визуальной части стандарта MPEG-4.

2.2. СЖАТИЕ ЗВУКОДАННЫХ

2.2.1. Эффект маскирования и психоакустическая модель слуха

Из-за повышенной чувствительности уха к искажениям сжатие звуковых сигналов оказывается более сложной задачей, чем видеокомпрессия. В то же время некоторые психофизиологические особенности слухового восприятия позволяют даже при существенной компрессии (6:1 и более) обеспечить «прозрачный» канал, т.е. канал, звучание выходного сигнала которого субъективно неотличимо от звучания сигнала источника. Для понимания этих особенностей познакомимся вкратце с механизмом слухового восприятия.

Известные нам из школьного курса анатомии барабанная перепонка и стремечко преобразуют звуки в механические вибрации и передают их на находящийся в улитке внутреннего уха кортиев орган — протяженную мембрану между двумя камерами, заполненными жидкостью. Звуки разной высоты возмущают различные участки мембраны, имеющие разные резонансные частоты, и через нервные окончания эти возбуждения передаются в мозг. Частотный диапазон восприятия простирается от 20 Гц до примерно 20 кГц, область наибольшей чувствительности лежит между 1 кГц и 5 кГц. Кривая чувствительности слуха в отсутствие громких звуков показана на рис. 2.8а. Как и во всякой системе с преобразованием из временной в частотную область, в слуховой системе имеется некоторая неопределенность в разделении двух событий: чем точнее известны характеристики сигнала во временной области, тем менее точно можно описать его спектральный состав, и наоборот. Несовершенное разделение по частоте приводит к тому, что ухо не в состоянии различить звуки с близкими частотами, этот эффект

BEST AVAILABLE COPY

The most effective method of reading the coefficient values is zigzag scanning from the upper left angle to the lower right angle (see Figure 2.7), comprising the steps of first counting all non-zero coefficients followed by reading zero coefficients. To encode the latter, the above-mentioned method of run-length coding is used that comprises assigning separate code words to each continuous sequence of identical symbols (zeroes in this case).

statistical probability for occurrence of different coefficient values is non-equal: some coefficients occur very frequently, others occur rarely. This aspect can be used for further reducing the number of bits, if shorter sequences of symbols are assigned to values occurring more frequently, and longer sequences of symbols are assigned to values occurring more rarely. This problem is solved by variable-length codes mentioned above.

Depending upon a picture subject and an object motion velocity, a fraction of redundancy to be removed varies, and the average number of bits per sample may vary in wide limits. A digital stream rate at an output of a compression encoder will wary within the same limits. At the same time, most communication channels are designed to transmit signals at constant rates. As usual in these cases, buffer memory helps. The buffer accumulates redundant bits at the moments (change of a subject, fast motion) critical for the encoders, and provides said bits to a channel during the periods of relative tranquility when the channel is underloaded with current data. There is provided a feedback mechanism as well – for the of the buffer overflow, or contrary, buffer underflow: by adaptively changing the quantization matrix coefficients, the buffer can control an incoming number of bits and a stream rate within certain limits, of course, due to respective increase or decrease in a quantization step and quantization noise associated therewith.

2.1.5 Some Current Digital Compression Standards

Effective methods for digitally compressing images, developed at the late eighties and described above, have given growth to a great number of processing schemes that combine different techniques and are directed to solution of different problems. A part of said schemes was adopted by standardization authorities and gained the status of international standards. Let us consider the most effective of them that became the basis for standards of the MPEG family.

In 1982 – 1990, the Joint Photographic Expert Group organized by ISO, IEC and ITU together developed and then adopted the standard with the same name for encoding still images [2.6]. The main problem stated was to reduce the amount of data necessary for transmission of a

BEST AVAILABLE COPY

high-quality picture (up to 0.25 bit/pixel for a "distinguishable" picture, 1 bit/pixel for "excellent" quality, and 4 bit/pixel for a picture "non-distinguishable" from the original one, at a starting quantization accuracy of 8 bit/pixel).

Many digital compression algorithms were tested during long studies, and the DCT was selected as the one providing the best quality at the minimum digital stream rate. The technical solutions included in the subsequent standards, namely, processing not original color signals but matrixed luminance signals and chrominance signals, and scaling the latter a half to the format "4:2:0", selecting the size of the DCT blocks to be 8 x 8 pixels, zigzag scanning the coefficients, separately quantizing constant components and other DCT coefficients, entropy encoding, were introduced first in the JPEG standard. Only motion compensation was lacking for understandable reasons.

The next standard of interest – H.261 – was developed in 1988-90 by the Consultative Committee or International Telephony and Telegraphy (CCITT) that was the precursor of the present ITU-T. The standard was designed for transmission of video images (video telephony, video conference) at the rates of n x 64 kbit/s, where n = 1...30, and the image format could not be higher that CIF. That standard already used the prediction encoding and two types of frames: I- and B-frames. The motion prediction was performed by comparison with a previous frame, and an absolute error minimum was taken as the match criterion. The original search window was a rectangular section with a size of 64 to 48 pixels, and the search could be performed by one of three methods: 1) complete searching the entire search area (a slow method); 2) sequential narrowing the window till the required match accuracy is attained; 3) hierarchical motion estimation that requires several sequences of versions of a reference frame and a current frame with progressively lower resolution, wherein the best match in the version with the lowest resolution is searched for and counted to the starting level by sequentially modifying the displacement vectors. Similar to the present standards, a macro block having the size of 16 x 16 pixels contained 4 blocks of luminance signal samples and one block per chrominance signal.

The H.261 and JPEG standards became the basis to develop MPEG-1.

The research in this direction continued, and in 1996 the ITU-T adopted the H.261 standard for the same applications, that was more appropriate for low-rate encoding. That standard supported formats from SQCIF to 16CIF (1408 x 1152 pixels), and dealt with all three

BEST AVAILABLE COPY

types of frames – I, P, and B, wherein the match search area was not limited, and the match accuracy was 0.5 pixel. The basic improvements as compared to H.261 were a non-limited range of variation of the displacement vectors (they could point beyond the picture area), arithmetical encoding of the quantized DCT coefficients, improved prediction (4 blocks of 8 x 8 vectors were predicted instead of one block having a size of 16 x 16 pixel), encoding PB-frames (the capability to encode sequential P- and B-frames together as one PB-frame). The H.262 codec exceeded the rate of 64 kbit/s and became a broadly used codec for low-rate encoding.

Table 2.1 contains several parameters of H.261 and H.263 standards for various picture formats.

Format	Number of luminance signal samples	Number of chrominance signal samples	H.261 standard	H.263 standard
SQCIF	128 x 96	64 x 88	optionally	supported
QCIF	176 x 144	88 x 72	supported	supported
CIF	352 x 288	176 x 144	optionally	optionally
4CIF	704 x 576	352 x 288	not supported	optionally
16CIF	1408 x 1152	704 x 576	not supported	optionally

The second version of the H.263 standard developed recently and named as H.263+ contains many additional functions that make this standard completely modern and effective: the intra-frame compression mode (filter unblocking) is improved; prediction is performed on the basis of a resampled reference picture; the low-resolution updating mode is introduced; reversible variable-length codes are used; a picture format and clock frequencies can be customized; the time, space and signal-to-noise scaling modes are introduced; the error resistance is improved due to structuring the slices.

Let us note that the H.263+ standard has been used in development of the visual part of the MPEG-4 standard.

Figure 2.7. Zigzag scanning of the digital cosine transform (DCT) coefficients

BEST AVAILABLE

OIPE
AUG 1 8 2006
PATENT & TRADEMARK OFFICE
IAP19

IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,996	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on August 15, 2006.

By: Tracey M. Dotter
Tracey M. Dotter

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. §1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the item of information listed on the enclosed Form 1449 is brought to the attention of the Examiner.

This statement should be considered because it is submitted either within three months of the filing date or before the first Office Action on the merits of the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

In accordance with 37 C.F.R. §1.98(a)(2), and the 5 August 2003 Official Gazette Notice, only a copy of each foreign document or other non-patent published document listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

OIPE
AUG 18 2006
PATENT & TRADEMARK OFFICE
IAP19

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. §1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP §609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: August 15, 2006

By: Erin M. Nichols
Erin M. Nichols
Reg. No. 57,125

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

OIP IAP19 AUG 18 2006 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	1998	Wenger, "Temporal Scalability Using P-pictures for Low-latency Applications", IEEE Workshop, December 1998, pp. 559-564.

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TEMPORAL SCALABILITY USING P-PICTURES FOR LOW-LATENCY APPLICATIONS

Stephan Wenger
stewe@cs.tu-berlin.de
Technische Universität Berlin
Sekr. FR 6-3, Franklinstr. 28-29
D-10587 Berlin, Germany
Tel.: +1-415-297-2063 Fax.: +49-30-314-25156

Abstract: Many of the newer video coding/compression standards, including MPEG-4 and H.263+ support some form of temporal scalability as part of their layered codec concept. This is generally realized by using bi-directionally predicted pictures (B-pictures), which use both an earlier and a subsequent P-picture as reference (anchor) pictures. This paper introduces a new form of temporal scalability employing only P-pictures. This mechanism results in improved real-time behavior (particularly lower latency) and a more flexible layering structure, at the cost of less efficient coding. The P-picture based temporal scalability mechanism is particularly useful for interactive multimedia communication on networks that offer several independent transport streams (possibly with different quality of service), but have sub-optimal real-time characteristics. Typical applications include both the Internet and some forms of mobile communication. The 1998 version of H.263 (known as H.263+ in both academia and industry) offers a mechanism to support P-picture scalability within the bit-stream through the Reference Picture Selection mode [1] [2]. Other video coding standards, including those of the MPEG family, require slight modifications to the decoder in order to support the proposed mechanism.

INTRODUCTION

The concept of a layered codec is well known [3] [4] and recent video coding standards have incorporated support for such mechanisms. A layered video codec produces a representation of the video sequence at various resolutions and qualities. This allows for compatible and robust representations. A particular representation can be obtained by decoding a particular subset of the coded data. Typically, each representation is referred to as a layer. In a hierarchical scalable coding scheme, each layer is coded with respect to lower layer representations. These layers can then be transmitted in separate data streams. The base layer is the only self-contained representation of the sequence and is usually low quality, low resolution, and low bit-rate. Enhancement layers code the difference information or error between the base layer and a higher quality or resolution representation of the video sequence. Three forms of scalability are widely used:

- Temporal enhancement layers increase the frame-rate of the reproduced video by coding additional pictures.
- Spatial enhancement layers raise the spatial resolution of the coded picture by coding difference information between an up-scaled reference layer picture and a higher resolution representation of the original picture. Decoding the base layer produces a low spatial resolution representation, while decoding both the base and spatial enhancement layers produces a higher resolution representation.
- Signal-to-noise-ratio enhancement layers (SNR scalability) allow the transmission of additional precision of DCT coefficients (and/or motion vectors), thus leading to a higher quality reproduced image and consequently to an increase of the SNR of that reproduced image.

All different enhancement layer mechanisms can be used in combination, although some restrictions exist in the current standards. In H.263+, for example, it is possible to use a spatial enhancement layer to increase the spatial resolution of the representation. Using this higher resolution representation, temporal scalability can be employed to produce a higher frame-rate. However, it is not permitted to use the pictures of a temporal enhancement layer as a reference for subsequent SNR or spatial enhancement layers.

This paper focuses on temporal scalability only. It introduces a new form of temporal scalability employing P-pictures. The following section describes this P-picture scalability and distinguishes it from usual B-picture scalability, using H.263+ as an example. Simulation results for both are presented next. The final section describes the applicability of P-picture scalability to other video coding standards implementing inter picture prediction.

THE CONCEPT OF P-PICTURE BASED TEMPORAL SCALABILITY

When implementing a layered codec based on H.263+, the common solution is to employ H.263+'s Temporal, Spatial, and SNR Scalability optional mode (Annex O). In this mode, temporal scalability is realized through bi-directionally predicted pictures (B-pictures). B-pictures use the previous and the subsequent P-picture (or I-picture) as their anchor pictures. Figure 1 illustrates a base layer that is decodable at 10 fps and an B-picture enhancement layer that is decodable at 30 fps.

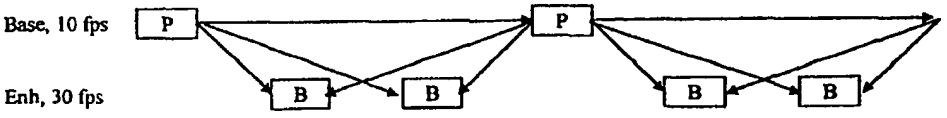


Figure 1: Base and temporal enhancement layer.

As stated above, H.263+ does not permit the B-pictures of the enhancement layer to be used as reference pictures for subsequent layers. Furthermore, with this mechanism it is necessary to decode two base layer pictures before it is possible to decode the B-pictures. Therefore, the overall latency of the base/enhancement layer combination and of the base layer itself are similar, although bits were spent in the enhancement layer to increase temporal resolution. The advantage of temporal scalability employing B-pictures is the comparably small size of B-pictures, leading to a high coding efficiency in the enhancement layer. This is in part due to the enhanced (bi-directional) prediction mechanisms for B-pictures. Additionally, a higher quantization step size is often used for the enhancement layer B-pictures compared to the base-layer P-pictures. This is possible because of the restriction that B-pictures not be used as references whatsoever. The result is high coding efficiency for the enhancement layer B-pictures and little noticeable impact on the reproduced video quality.

The proposed P-picture based temporal scalability mechanism is illustrated in Figure 2. The base layer is a usual sequence of P or I-pictures, as in the example above. The enhancement layer, however, consists only of P-pictures. In addition, the enhancement layer now uses previous pictures of the enhancement layer, as well as base layer pictures, as reference pictures. Two scenarios are shown:

- a base layer that is decodable at 15 fps with an enhancement layer that is decodable at 30 fps, and
- a base layer that is decodable at 10 fps with an enhancement layer that is decodable at 30 fps.

Note that the proposed mechanism can also employ a higher quantizer step-size in the enhancement layer P-pictures to increase coding efficiency with little noticeable impact. This is a particularly useful technique for the last picture of each enhancement layer sub-sequence.

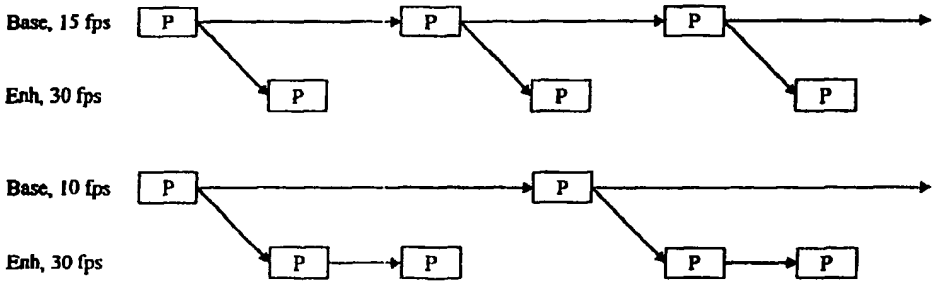


Figure 2: Principle of P-picture temporal scalability

Employing P-pictures for temporal scalability is not an explicitly defined optional mode in H.263+, or in any of the other popular video coding standards. However, in H.263+ it is possible to use the Reference Picture Selection optional mode

(Annex N) without a back channel to generate a standard-compliant bit stream containing scalable P-picture information. This optional mode allows, for each predicted picture, one of several recently transmitted pictures to be selected for use as the reference picture, as opposed to simply allowing the most recent picture to be used as the reference picture. This is signaled in the picture header by coding the temporal reference of the selected reference picture. An H.263+ decoder supporting this optional mode must store several previous reference pictures along with their associated temporal references.

The proposed P-picture based temporal scalability mechanism provides several advantages over the traditional B-picture based temporal scalability mechanism. First, the latency is reduced when spending bits for the enhancement layer. Since a P-picture can be decoded without backward prediction, the enhancement layer P-pictures can be decoded as soon as they are transmitted. At an enhancement layer frame rate of 30 fps, and a base layer frame rate of 10 fps, the overall latency can be 66 milliseconds shorter than using B-frame scalability (33 ms instead of 100 ms). Second, P-frames can be used as reference frames for subsequent enhancement layers. This provides more flexibility in the design of a layered codec. Third, error resilience is improved, as it is by using B-picture based temporal scalability. Any damage to an enhancement layer P-picture will not affect prediction starting from the next reference layer P-picture. The main disadvantage is the inferior coding efficiency of P-pictures relative to B-pictures. This reduced coding efficiency in the enhancement layer results in a slight increase in bit rate.

SIMULATION RESULTS

Using the University of British Columbia's H.263+ reference codec and our implementation of the Reference Picture Selection optional mode (Annex N), both B-picture based temporal scalability (using Annex O) and P-picture based temporal scalability (using Annex N) are simulated. A base layer decodable at 10 fps and an enhancement layer decodable at 30 fps are generated. Fixed quantizer step sizes of 13 and 16 are used for the base and enhancement layers respectively. The input sequences Foreman, Coastguard and Paris, at QCIF resolution, are employed. Several improved coding-efficiency oriented optional modes, Advanced Prediction mode (Annex F), Advanced Intra coding mode (Annex I), Deblocking Filter mode (Annex J) and Modified Quantization mode (Annex T) are used. This configuration produces representations at bit rates typical for Internet scenarios employing layered codecs. The resulting picture quality is far beyond what is achievable using current Internet applications (e.g. the popular Mbone tools).

The picture quality for B-picture based scalability and the proposed P-picture based scalability is comparable, with a PSNR difference of less than 0.1 dB. This result is expected as the same quantizer values are used for both sets of

simulations. Therefore, we compare the different layering mechanisms based on coding efficiency, using the data in Table 1. The first column shows the bit rate for a non-layered representation decodable at 10 fps. The second column shows the bit rate for a non-layered representation decodable at 30 fps. This bit-rate can be considered as the optimum for full temporal resolution (i.e. 30 fps) P-picture based coding. The third and fourth columns show the bit rate of the enhancement layers for layered representations decodable at 30 fps using B-picture based temporal scalability and P-picture based temporal scalability respectively. The fifth and sixth columns show the total bit rate of the base and enhancement layers combined for layered representations decodable at 30 fps using B-picture based temporal scalability and P-picture based temporal scalability respectively. This data provides an indication of the network load imposed by the different mechanisms. Of particular interest are columns 4 and 6, which illustrate the bit rate overhead of P-picture based temporal scalability as compared to B-picture based temporal scalability.

Sequence	Base 10 fps p--p--	Base 30 fps pppppp	Enhance -BB-BB-	Enhance -PP-PP-	Total pBBpBB	Total pPPpPP
Paris	58.6	106.6	25.3	60.5 139%	83.9	119.1 42%
Foreman	43.8	108.5	29.9	45.7 52%	73.7	89.5 21%
Coastguard	60.5	113.0	18.7	25.9 38%	79.2	86.4 9%

Picture types: p: base layer P, P: enhancement layer PB: enhancement-layer B

Table 1: Performance of the P-picture temporal scalability scheme relative to B-picture scalability

The data shows a sequence-dependent total bit rate increase of 9% to 42% when P-picture based temporal scalability is employed, as compared to B-picture based temporal scalability. The associated enhancement layer bit rate increase is 38% to 139%. For many applications, this increase in bit rate is tolerable, given that the benefits of a scalable bit stream are provided with significantly reduced latency using the proposed P-picture based temporal scalability mechanism.

USING P-PICTURE BASED TEMPORAL SCALABILITY OUTSIDE OF H.263+

H.263+ offers a mechanism for P-picture based temporal scalability through the Reference Picture Selection optional mode. The transport layer can then create a single P-picture based scalable bit stream from multiple transport streams (each of them conveying one layer) by using the temporal reference as the ordering criterion and concatenating the coded pictures. Unfortunately, other current video coding standards lack such a mechanism. This section provides a brief overview of how

to circumvent such limitations and implement a P-picture based temporal scalability mechanism in a non-H.263+ framework. In particular, the MPEG-family of ISO/IEC standards and ITU-T H.261 and H.263 version 1 are to be mentioned.

Using these standards, it is impossible to generate a standard compliant data stream that combines a base layer and a P-picture based temporal scalability layer without performing a complete decoding and re-encoding of the video sequence. However, it is possible to modify a decoder such that not only the most recently decoded P-picture is stored but also several recently decoded P-pictures are stored in a queue. The decoder can then use these stored P-pictures in the following manner:

- If the current picture belongs to the base layer, i.e. if it was conveyed over a base layer virtual transport channel, then use the most recently decoded base layer P-picture as a reference. This picture can easily be flagged in the decoded P-picture queue.
- If the current incoming picture belongs to the P-picture enhancement layer, i.e. if it was conveyed over an enhancement layer virtual transport channel, then use the most recently decoded P-picture (from any available source channel) as the reference.

The corresponding encoder operates in a similar manner:

- The base layer is coded in the usual manner, performing inter-picture prediction based on reference pictures from the low temporal resolution representation only.
- The enhancement layer is coded using the most recently coded P-picture as a reference.

We have implemented such a mechanism for a real-time commercial H.261 codec. The results are comparable to those presented above for the H.263+ case.

REFERENCES

[1] ITU-T Recommendation H.263 (1998): Video Coding for Low Bit Rate Communication, ITU-T, 1998

[2] G. Cote, B. Erol, M. Gallant, F. Kossentini: "H.263+: Video Coding for Low Bit Rates", IEEE T-CSVT, 9/1998

[3] M. K. Uz, M. Vetterli, and D. J. LeGall. Interpolative Multiresolution Coding of Advanced Television with Compatible Subchannels. IEEE Transactions on Circuits and Systems for Video Technology. Volume 1. Pp 86-99. March 1991.

[4] K. Ramchandran, A. Ortega, M. K. Uz, M. Vetterli. Multiresolution Broadcast for Digital HDTV using Joint Source-Channel Coding. IEEE Journal of Selected Areas in Communications. Special issue on HDTV and Digital Video Communications. Volume 11. pp 6-23. 1993.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☑ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

OIPE IAP43
NOV 24 2006
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,996	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on November 21, 2006.

By: Rennae Johnson
Rennae Johnson

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. §1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the item of information listed on the enclosed Form 1449 is brought to the attention of the Examiner.

This statement should be considered because it is submitted either within three months of the filing date or before the first Office Action on the merits of the above-identified application. Accordingly, no fee is due for consideration of the item listed on the enclosed Form 1449.

In accordance with 37 C.F.R. §1.98(a)(2), and the 5 August 2003 Official Gazette Notice, only a copy of each foreign document or other non-patent published document listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached document do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

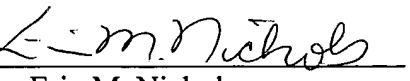
No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. §1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the item listed is respectfully requested. Pursuant to the provisions of MPEP §609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: November 20, 2006

By: 
Erin M. Nichols
Reg. No. 57,125

OIPE IAP43 NOV 24 2006 PATENT & TRADEMARK OFFICE

FORM 1449			
INFORMATION DISCLOSURE STATEMENT		Docket Number: TPO.004.C1	Application Number: 11/338,996
IN AN APPLICATION		Applicant: HANNUKSELA	
(Use several sheets if necessary)		Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	RU 2 123 769	20.12.1998	Russia			Abstract Only	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

(19) **RU** (11) **2 123 769** (13) **C1**

(51) МПК[6] **H 04 N 9/79**

РОССИЙСКОЕ АГЕНТСТВО
ПО ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ

(21), (22) Заявка: 94046325/09, 11.04.1994

(30) Приоритет: 09.04.1993 JP P5-105943

(46) Дата публикации: 20.12.1998

(56) Ссылки: US 4723175 A, 02.02.88. DE 3809207 A, 29.09.88. JP 3-217183 A, 24.09.91. JP 3-252284 A, 11.11.91. SU 1434563 A, 30.10.88.

(86) Заявка PCT:
JP 94/00610 (11.04.94)

(71) Заявитель:
Сони Корпорейшн (JP)

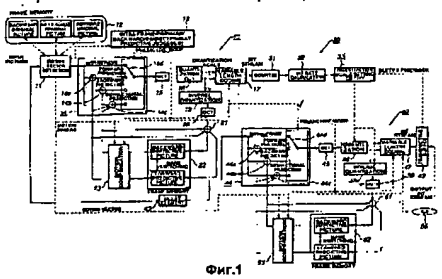
(72) Изобретатель: Цуеси Ода (JP)

(73) Патентообладатель:
Сони Корпорейшн (JP)

(54) СПОСОБЫ И УСТРОЙСТВА КОДИРОВАНИЯ ИЗОБРАЖЕНИЙ И НОСИТЕЛИ ИНФОРМАЦИИ ДЛЯ ЗАПИСИ ИЗОБРАЖЕНИЙ

(57) Реферат:

Электронная схема и последовательность операций, позволяющие кодировать входной видеосигнал путем выполнения, например, прогнозирующего кодирования, ДКП обработки (Дискретное Косинусное Преобразование), квантования данных с фиксированным шагом квантования, неравномерного кодирования (с переменной длиной) для вырабатывания первых кодированных данных и последующего определения (начисления) количества кодов, отводимых (задаваемых) для каждого кадра или каждой группы изображений ГИ на основе учета количества данных в каждый заранее задаваемый отрезок времени, например за время длительности каждого кадра или каждой СОР в объеме первых кодированных данных и общего количества данных, пригодных для кодирования входного видеосигнала в каждый заранее задаваемый отрезок времени на основе назначаемого количества кодов, для выработки вторых кодированных данных. Таким образом осуществляется кодирование с переменной скоростью, когда скорость кодирования изменяется в каждый заранее задаваемый отрезок времени. В результате, если даже имеет место последовательность изображений (кадров) сложного рисунка, нет возможности обеспечить столь же большое значение шага квантования для этих изображений, как в обычных устройствах. Благодаря этому можно достигнуть однородного изображения высокого качества в целом в самых различных случаях, что является достигаемым техническим результатом. Кроме того, поскольку вторые кодированные данные имеют переменную скорость, то в случае, когда такие данные регистрируются на носителе записанного изображения, ограниченный объем памяти может быть использован более эффективно, а время записи изображения может быть продлено. Наконец, данные изображения, имеющего высокое качество и однородность во всех возможных случаях, могут быть воспроизведены с носителя записанного изображения. 9 с. и 18 з.п.ф-лы, 12 ил.



Фиг.1

RU 2123769 C1

RU 2123769 C1

(19) **RU** (11) **2 123 769** (13) **C1**

(51) Int. Cl.[6] **H 04 N 9/79**

RUSSIAN AGENCY
FOR PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION

(21), (22) Application: 94046325/09, 11.04.1994

(30) Priority: 09.04.1993 JP P5-105943

(46) Date of publication: 20.12.1998

(86) PCT application:
JP 94/00610 (11.04.94)

(71) Applicant:
Soni Korporejshn (JP)

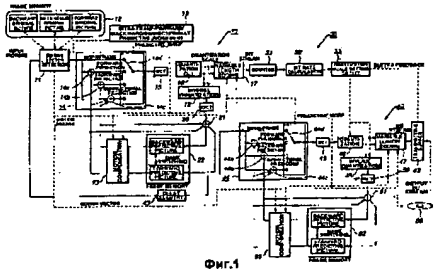
(72) Inventor: Tsuesi Oda (JP)

(73) Proprietor:
Soni Korporejshn (JP)

(54) METHOD AND DEVICE FOR ENCODING IMAGES AND INFORMATION MEDIUM FOR STORING IMAGES

(57) Abstract:

FIELD: computer engineering, in particular, for encoding of input signal by running prediction encoding, discrete cosine transform, data sampling with constant quantization step, non-uniform encoding (with alternating length). SUBSTANCE: method involves encoding of first data group and subsequent detection (assigning) code length for allocation (setting) for each frame or each image group using data length in each given time interval, for example, for duration of each frame in first encoded data group and total length of data available for encoding of input video signal in each given time interval using assigned number of codes for generation of second group of encoded data. This results in possibility of alternating length encoding, when encoding rate alters in any given time interval. So, even if sequence of images (frames) for complex picture is given, there is not way to ensure large quantization step for these images as in state-of-the-art devices. This provides possibility to generate uniform image of high quality in various cases. In addition since second group of encoded data has alternating rate, then if such data are recorded onto information medium, a restricted size of buffer memory may be used with greater efficiency and information storage duration may be extended. Finally, image data of high quality and uniformity in all possible cases can be read from medium which stores this image. EFFECT: increased image quality, increased efficiency of buffer memory usage. 28 cl, 12 dwg



Фиг.1

RU 2123769 C1

RU 2123769 C1

Область применения.

Настоящее изобретение относится к способу кодирования подвижного изображения, к устройству кодирования подвижного изображения и к носителю информации в виде записанного подвижного изображения, а более конкретно оно относится к способу и устройству кодирования подвижного изображения, к носителю записанного подвижного изображения, который используется в системе для кодирования и последующего хранения, к видеосигналу подвижного изображения, к регистрации кодированного сигнала на носителе информации, таком как, например, оптический диск, магнитный диск или магнитная лента и т.п., а также к системам для передачи видеосигнала подвижного изображения через передающую среду.

Предпосылки создания изобретения.

До последнего времени в системах передачи видеосигнала подвижного изображения в удаленные точки, как, например, в телевизионных системах передачи с конференцией, в телевизионных телефонных системах и т.п., или в системах регистрации видеосигнала на таких носителях, как оптический диск, магнитный диск или магнитная лента и т.п., или при воспроизведении записанного видеосигнала подвижного изображения, была широко распространена так называемая схема эффективного кодирования при более эффективном использовании трассы передачи видеосигнала (или носителя записи изображения), когда за счет использования корреляции между строками или корреляции между кадрами видеосигнала снижается избыточность по пространственной оси и по временной оси, и передается только наиболее существенная информация, что увеличивает эффективность передачи.

Например, при обработке в виде кодирования в направлении пространственной оси (далее в тексте - внутрикадровое кодирование) корреляция между строками видеосигнала используется так, как показано на фиг. 7. При попытке передать соответствующие изображения PC1, PC2, PC3,... представляющие собой в моменты времени t1, t2, t3,... подвижные изображения, данные изображения, подлежащие обработке и передаче, подвергаются одномерному кодированию, например, в рамках одной и той же строки, или же изображение разбивается, например, на несколько блоков, что позволяет подвергнуть данные изображения в соответствующих блоках двумерному кодированию с тем, чтобы осуществить сжатие данных, что увеличивает эффективность передачи.

Кроме того, при обработке кодированием в направлении оси времени (далее также - внутрикадровое кодирование) внутрикадровая корреляция видеосигнала используется для определения (с помощью так называемого прогнозирующего кодирования) данных изображения PC12, PC23,... имеющих недостаток элементов (так называемых прогнозированных ошибок) в каждом из пикселей данных изображений, общих для смежных изображений PC1, PC2 и PC3, в последовательности этих данных PC12, PC23,... подлежащих передаче, осуществляя, таким образом, сжатие данных, а следовательно, и увеличивая эффективность передачи.

Таким образом, по сравнению со случаем, когда передаются все данные изображений PC1, PC2, PC3,... в данном случае видеосигнал может быть передан на основе значительно меньшего количества данных.

Кроме того, в процессе прогнозирующего кодирования во время описанного выше внутрикадрового кодирования применяется также прогнозируемая компенсация движения, например, в объеме массивов, состоящих из макроблоков, с тем, чтобы еще более увеличить эффективность. Например, когда персонаж в центральной части изображения движется, такое движение объекта на изображении обнаруживается с целью коррекции положения данных изображения, используемых для прогнозирования в предыдущем изображении путем выполнения прогнозируемого кодирования в предыдущем изображении, что также дает возможность увеличить эффективность кодирования. Однако, даже когда применяется такое прогнозирование на основе компенсации движения, большое количество данных по сравнению с той частью изображения, где объект движется и появляется на заднем фоне, должно быть передано. Ввиду этого, не только компенсация движения в описанном выше направлении вперед, но также и компенсация движения в направлении назад или даже в обоих направлениях - вперед и назад - используется совместно, что дает возможность еще более увеличить эффективность кодирования.

Фактически, как показано на фиг. 8, в макроблоках данных кадров F0, F1, F2, F3 нулевого, первого, второго, третьего,... кадров видеосигнала подвижного изображения, подлежащего передаче, и в случае, когда имеют место изменения в изображении, что соответственно указано векторами движения x0, x1, x2, x3, ... последовательно между кадрами, устройство на передающей стороне обозначает кадры в заранее заданном количестве интервалов (например, исключая каждый следующий кадр), т.е. второй, четвертый,... как кадры интерполяции для так называемой заранее задаваемой кадровой интерполяции, как показано на фиг. 8, с целью формирования данных для передачи в виде интерполированных кадров F2X, F4X. . . . Далее, в отношении неинтерполированных кадров, устройство на передающей стороне выполняет заранее задаваемое колирование кадров F1, F2, F3,... для формирования подлежащих передаче данных - неинтерполированных кадров F1X, F3X,....

Например, разность SP2 (прогнозируемая ошибка) между данными кадров с компенсацией движения F3 и F2, разность SP4 между данными кадров с компенсацией движения F1 и F2 и разность между данными кадров, полученных в результате интерполяции данных кадров с компенсацией движения F1 и FT3, а также данные кадра F2 соответственно определяются в массивах макроблоков для сравнения разности (данных) SP1 данных кадра F2 и этих разностей. Затем данные, в составе которых

RU 2123769 C1

RU 2123769 C1

имеется минимальное количество данных, сформированных на основе SP1, SP4, выводятся на передачу как интерполированные данные F2X в массивах макроблоков. Аналогично интерполированные для передачи данные F4X, ... генерируются на основе соответствующих кадров интерполяции. Далее, данные неинтерполированных кадров F1, F3,... подвергаются обработке ДКП (дискретное косинусное преобразование) и неравномерному кодированию для формирования подлежащих передаче данных неинтерполированных кадров F1X, F3X, ...

Данные неинтерполированных кадров F1X, F3X,... и данные интерполированных кадров F2X, F4X, . . . передаются в устройство на приемной стороне совместно с векторами движения x0, x1, x3,...

С другой стороны, устройство на приемной стороне выполняет обработку декодированием, соответствующим обработке кодированием на предающей стороне, перед передачей данных (передаваемые данные неинтерполированных кадров F1X, F3X, . . . передаваемые интерполированные данные кадров F2X, F4X,... данные векторов движения x0, x1, x3) с тем, чтобы воспроизвести данные кадров F0, F1, F2, F3, . .. В результате компенсации движения выполняются не только в направлении вперед, но и в направлении назад или даже в обоих направлениях - вперед и назад, что дает возможность увеличить еще более эффективность кодирования.

Ниже приводится описание устройств кодирования и декодирования, имеющих описанные выше функции.

Устройство кодирования содержит, как показано на фиг. 9, схему 61 предварительной обработки, разделяющую видеосигнал BC на сигнал яркости и цветоразностный сигнал, преобразовательные схемы 62a и 63б аналог/цифра (далее в тексте А/Ц схемы) для преобразования сигнала яркости и цветоразностного сигнала, соответственно поступающих со схемы 61, в цифровые сигналы, ЗУ кадровой памяти 63 для хранения данных яркости и цветоразностных данных (далее в тексте - данные изображения), поступающих со схем А-Ц 62a и 62б, схему 64 преобразования, схему 64 преобразования формата, служащую также для считывания данных из ЗУ 63 кадровой памяти в соответствии с блочным форматом, и кодирующее устройство (кодер) 65 для эффективного кодирования блока, поступающего со схемы 64 преобразования формата, в данные изображения.

В процессе работы схема 61 предварительной обработки разделяет входной видеосигнал BC на сигналы яркости и цветоразностный. А/Ц схемы преобразования 62a и 62б соответственно преобразуют сигнал яркости и цветоразностный в данные яркости и цветоразностные данные группами по 8 бит каждая. ЗУ 63 хранит эти данные.

Схема 64 преобразования формата считывает в соответствии с блочным форматом, данные изображения (данные яркости и цветоразностные данные, хранящиеся в ЗУ 63 кадровой памяти). Кодер 65 кодирует данные изображения, считывая их таким образом с заданной эффективностью кодирования в выходной поток бит.

Этот поток бит направляется в устройство 80 декодирования изобретения через передающую среду 70, которой может быть либо трасса распространения сигнала, либо носитель записи изображения, например оптический диск, магнитный диск или магнитная лента и т.п.

Это устройство декодирования 80 содержит, как показано на фиг. 9, декодер 81, соответствующий кодеру 65, схему преобразования формата 82, служащую для преобразования данных изображения, воспроизведенных декодером 81, в формат кадра, ЗУ 83 кадровой памяти для хранения данных изображения, подаваемых со схемы 82 преобразования формата, Ц/А преобразовательные схемы 84a, 84б для преобразования данных яркости и цветоразностных данных, которые были считаны из ЗУ 83 кадровой памяти, в аналоговые сигналы, и послеобработочная схема 85 для смешения сигнала яркости и цветоразностного сигнала, подаваемых со схем 84a, 84б Ц/А преобразования, в результате чего формируется выходной видеосигнал.

Декодер 81 декодирует поток бит, выполняя операцию декодирования, соответствующую эффективному кодированию в кодере 65, для воспроизведения данных изображения в блочном формате. Схема 82 преобразует эти данные изображения и в формате кадра передает их на хранение в ЗУ 83 кадровой памяти.

Ц/А преобразовательные схемы 84a, 84б соответственно преобразуют данные яркости и цветоразностные данные, считанные из ЗУ 83 кадровой памяти, в сигнал яркости и в цветоразностный сигнал. Послеобработочная схема 81 смешивает эти два сигнала, формируя таким образом выходной видеосигнал.

Фактически послеобработочная схема 61 и Ц/А преобразовательные схемы 62a, 62б преобразуют сигнал яркости и цветоразностный сигнал в цифровой сигнал так же, как это делается и описано выше для сокращения количества данных, так что количество пикселей становится равным половине пикселей сигнала яркости в верхнем и нижнем направлениях, а также в левом и правом направлениях по отношению к сигналу яркости при дальнейшем осуществлении мультиплексирования по оси времени вдобавок к выдаче данных яркости и цветоразностных данных, полученных таким образом, в ЗУ 63 кадровой памяти.

Из ЗУ кадровой памяти 63 данные яркости и цветоразностные данные считываются в соответствии с блочным форматом, как описано выше, а именно, например, данные изображения одного кадра разделяются на N секций, как показано на фиг. 10A. Каждая секция формируется так, чтобы в ней содержалось M макроблоков, как показано на фиг. 10B. Каждый макроблок составляется из данных яркости в виде четырех блоков яркости Y1, Y2, Y3, Y4, каждый из которых состоит из 8 x 8 пикселей, в верхнем и нижнем, а также в левом и правом

RU 2123769 C1

RU 2123769 C1

направлениях, и из цветоразностных данных Сб, Ср цветных блоков, состоящих из 8 х 8 пикселей, соответствующих по размерам блокам яркости. Из ЗУ 63 кадровой памяти данные яркости и цветоразностные данные считываются так, что данные изображения затем располагаются последовательно в массивах макроблоков в пределах секций и последовательно в порядке Y1, Y2, Y3, Y4, Сб, Ср внутри макроблока. Данные изображения, считанные таким образом в соответствии с блочным форматом, затем пересылаются в кодер 65.

Кодер 65 содержит схему 101 определения вектора движения, как показано на фиг. 11. Эта схема 101 обнаруживает (в массивах макроблоков) вектор движения данных изображения, передаваемых дополнительно в соответствии с блочным форматом. Схема 101 обнаружения вектора движения обнаруживает в массивах макроблоков вектор движения текущего изображения как опорного на основе исходных изображений в направлении вперед/назад, хранящихся в ЗУ 63 кадровой памяти. Обнаружение осуществляется так, что минимальный вектор сумм абсолютных значений разностей между кадрами в массивах макроблоков становится соответствующим вектору движения. Определенный таким образом вектор движения выдается на схему 113 компенсации движения и далее, а внутрикадровые разности в массивах макроблоков выдаются на схему внутрикадрового прогнозирования вперед/назад/в обоих направлениях 103.

Эта схема 103 определяет прогнозируемый вид образцового блока с тем, чтобы на основе значения его параметра управлять схемой 104 прогнозируемого кодирования, выполняя переключения внутрикадрового прогнозирования вперед/назад/в обоих направлениях в массивах макроблоков. Схема 104 прогнозирования кодирования содержит схемы суммирования 104а, 104б, 104в и селективный переключатель 104г, эта схема в том случае, когда прогнозируемый режим кодирования оказывается внутрикадровым режимом, она отбирает разности между всеми пикселями (далее в тексте - разностные данные) входных данных изображения относительно соответствующих прогнозируемых изображений, а затем выдает отобранные данные на схему 105 ДКП (Дискретное Косинусное Преобразование).

Схема 105 выполняет ДКП-обработку данных входного изображения или разностных данных в массивах блоков, используя двумерную корреляцию видеосигнала для определения данных коэффициента и последующей передачи его на схему 106 квантования.

Схема квантования 106 квантует данные коэффициента, соблюдая размер шага квантования (шкала квантования), определяемый для каждого макроблока или секции с тем, чтобы затем выдать полученные таким образом данные на схему 107 неравномерного кодирования (с переменной длиной) (далее по тексту - НК) и на инверсную схему 108 квантования. При определении шага квантования необходимо следить за тем, чтобы величина этого шага не стала причиной разрыва упорядоченности в буферной памяти 109 передачи, которая будет описана ниже, вследствие возможного образования обратной связи по остаточным данным этого буфера 109. Значение этого шага квантования также передается на схему НК 107 и на инверсную схему квантования 108.

Схема НК 107 выполняет неравномерное кодирование квантованных данных, имеющих определенный шаг, в прогнозируемом режиме с учетом вектора движения, передавая затем все эти параметры в буфер 109 в качестве данных, подлежащих передаче.

Буфер 109 передачи временно хранит данные, подлежащие передаче с тем, чтобы в определенный момент считать их с заранее заданной скоростью чтения битов и упорядоченно передать их на выход в виде потока битов, а также передает сигнал управления квантованием в массивах макроблоков обратно на схему 106 квантования в соответствии с количеством остаточных данных, остающихся в буферной памяти, и для управления значением шага квантования. Таким образом, буфер передачи 109 регулирует количество данных, формируемых в виде потока битов для поддержания надлежащего остаточного количества данных в памяти (остаточная емкость) с тем, чтобы не допустить переполнения буфера и недостатка данных. Например, когда количество остаточных данных в буфере передачи 109 увеличивается до допустимого верхнего предела, буфер увеличивает шаг квантования схемы 108 с помощью управляющего сигнала, снижая таким образом количество квантуемых данных. С другой стороны, когда количество остаточных данных в буфере передачи 109 уменьшается до нижнего допустимого предела, буфер 109 передачи уменьшает с помощью сигнала управления квантованием размер шага квантования схемы квантования 106, увеличивая тем самым количество квантуемых данных.

Таким путем поток битов с выхода буфера 109 подается на блок 80 декодирования изображения через среду передачи 70, которая может представлять собой либо трассу распространения сигналов в эфире, либо носитель записанного изображения, например оптический диск, магнитный диск или магнитную ленту, передача производится с заранее заданной скоростью следования битов, что было описано выше.

С другой стороны, схема 108 инверсного квантования в инверсном режиме квантует уже квантованные данные, поступающие со схемы квантования 106 с тем, чтобы воспроизвести коэффициентные данные (при этом добавляются искажения, вызываемые квантованием), соответствующие выходным данным описанной выше схемы 105 ДКП, и выдать затем эти данные на схему 110 Инверсного Дискретного Косинусного Преобразования (далее по тексту - ИДКП). Схема 110 и ДКП выполняет ИДКП-обработку коэффициентных данных для воспроизведения данных изображения, соответствующих входным данным изображения во внутри-кадровом режиме кодирования, а также воспроизвести разностные данные, соответствующие выходным данным схемы 104 в режиме

RU 2123769 C1

RU 2123769 C1

прогнозирования вперед/назад/в обоих направлениях, передавая их затем на схему 111 суммирования.

Когда режим кодирования является прогнозирующим режимом вперед/назад/в обоих направлениях, на схему суммирования 111 подаются прогнозированные данные изображения с компенсацией движения со схемы 113 компенсации движения, что будет описано ниже, для суммирования прогнозированных данных изображения с компенсацией движения и разностных данных, тем самым воспроизводя данные изображения, соответствующие входным данным изображения.

Воспроизведенные таким образом данные изображения хранятся в кадровой памяти 112. Фактически, схема 108 инверсного квантования и схема суммирования 111 составляют схему местного декодирования, которая локально декодирует квантованные данные с выхода схемы 106 квантования с тем, чтобы затем записать декодированное изображение, получаемое таким образом, в кадровую память в качестве вперед или назад прогнозированного изображения. Общая кадровая память состоит из множества более мелких ячеек памяти. Здесь осуществляется переключение банка кадровой памяти. В соответствии с изображением, подлежащим кодированию, на выход выдается одиночный кадр в качестве вперед или назад прогнозированных данных изображения. Более того, в случае прогнозирования в обоих направлениях, вперед и назад прогнозируемые данные изображения, например, усредняются и выдаются на выход. Эти прогнозируемые данные изображения представляют полностью такие же изображения, как и изображения, воспроизведенные декодером 81, что будет описано ниже. Следующее изображение, подлежащее обработке, подвергается прогнозируемому кодированию вперед/назад/в обоих направлениях на основе этого прогнозированного изображения.

Так, данные изображения, считанные из кадровой памяти 112, направляются в схему 113 компенсации движения. Эта схема 113 выполняет компенсацию движения прогнозируемых данных изображения на основе вектора движения, а затем передает прогнозированные данные изображения с компенсацией движения на схему 104 прогнозируемого кодирования и на суммирующую схему 111.

Далее рассматривается декодер 81.

На вход декодера 81 поток битов подается через передающую среду 70, а затем этот поток подается на схему 202 Инверсного Кодирования с Переменной Длиной (ИКПД) через приемный буфер 201. Схема 202 ИКПД воспроизводит квантованные данные, вектор движения, прогнозируемый режим и шаг квантования из потока бит. Эти квантованные данные и размер шага квантования передаются в схему 203 инверсного квантования. Вектор движения подается на схему 207 компенсации движения, а режим прогнозирования устанавливается в суммирующей схеме 205.

Схема 203 инверсного квантования и суммирующая схема 205 совместно работают аналогично схеме локального декодирования кодера 61, а ЗУ кадровой памяти 206 и схема 207 компенсации движения соответственно выполняют те же функции, что и кадровая память 112 и схема 113 компенсации движения кодера 61. Декодирование осуществляется на основе квантованных данных, вектора движения, режима прогнозирования и шага квантования. В результате данные воспроизведения изображения выдаются с выхода схемы суммирования 205.

Как упоминалось выше, в обычных устройствах такого типа скорость кодирования потока бит, генерируемого кодером 65, устанавливается фиксированной в соответствии со скоростью передачи через передающую среду 70. При таком ограничении количество генерируемых данных, т.е. шаг квантования в схеме 106 кодера 65, находится под контролем. Другими словами, например, управление осуществлялось таким образом, что когда идет последовательность сложных изображений, шаг квантования задается увеличенным для сокращения количества генерируемых данных, тогда как при наличии последовательности более простых изображений шаг квантования уменьшается для увеличения генерируемых данных с тем, чтобы буфер 109 не переполнялся или его содержимое не было ниже установленного предела, благодаря чему поддерживается фиксированная скорость передачи.

Соответственно в обычных (известных) устройствах при наличии последовательности сложных изображений шаг квантования увеличивается, и при этом ухудшается качество изображения, тогда как, если имеет место последовательность простых изображений, то шаг квантования уменьшается. Как результат этого, одинакового качества изображения во всех возможных случаях получить не удается.

Кроме того, в случае регистрации потока бит на носителе регистрируемого изображения ограниченной емкости (по количеству данных) во избежание значительного ухудшения качества изображения относительно сложных изображений для всех случаев устанавливают высокую фиксированную скорость, которая уже не сказывается на качестве сложных изображений, но при этом уменьшается время регистрации.

Сущность изобретения.

Для решения изложенных выше проблем первый способ кодирования изображения согласно настоящему изобретению включает операции кодирования по меньшей мере части входного видеосигнала для генерации первых кодированных данных, определения скорости кодирования в каждый заранее заданный момент на основе количества данных в каждый заранее заданный момент существования первых кодированных данных и общего количества используемых данных и кодирования входного видеосигнала в каждый заранее заданный момент на основе скорости кодирования с тем, чтобы генерировать вторые кодированные данные.

Второй способ кодирования изображений согласно настоящему изобретению отличается тем, что в первом способе кодирования изображения по меньшей мере часть входного видеосигнала квантуется с фиксированным шагом квантования для

RU 2123769 C1

RU 2123769 C1

генерации первых кодированных данных.

Третий способ кодирования изображений согласно настоящему изобретению отличается тем, что включает операции заранее определяемого прогнозируемого кодирования или заранее задаваемого преобразующего кодирования по меньшей мере части входного видеосигнала для генерации первых коэффициентов данных, квантования этих первых коэффициентов данных с фиксированным шагом для генерации (выработки) первых квантованных данных, что позволяет подвергнуть первые квантованные данные неравномерному кодированию (с переменной длиной) для формирования первого потока бит, определения скорости кодирования в каждый заранее заданный момент времени на основе количества данных в первом потоке бит и общего количества пригодных к использованию данных, выполнения заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразующего кодирования входного видеосигнала для генерации вторых коэффициентов данных, квантования этих вторых коэффициентов данных с шагом квантования, выбираемым на основе скорости кодирования в каждый заранее заданный момент времени, для генерации вторых квантованных данных и разрешения вторым квантованным данным подвергнуться неравномерному кодированию (с переменной длиной) для формирования второго потока бит.

Третий способ кодирования изображений согласно настоящему изобретению отличается тем, что в первом способе кодирования изображений общее количество пригодных для использования данных пропорционально распределяется в каждый заранее заданный момент времени, благодаря чему определяется скорость кодирования в каждый заранее заданный момент времени.

Пятый способ кодирования изображения согласно настоящему изобретению отличается тем, что при использовании четвертого способа кодирования изображений скорость кодирования определяется в каждом кадре на основе количества данных в каждом кадре первого потока бит и общего количества пригодных к использованию данных.

Шестой способ кодирования изображений согласно настоящему изобретению отличается тем, что при осуществлении четвертого способа кодирования изображений скорость кодирования определяется в каждой Группе Изображений (GOP) на основе количества данных из состава по меньшей мере части каждой GOP, состоящей из множества кадров из первого потока бит и общего количества пригодных к использованию данных.

Седьмой способ кодирования изображений согласно настоящему изобретению отличается тем, что при использовании шестого способа скорость кодирования в каждой ГИ определяется на основе количества данных, относящихся к изображению, закодированному внутри кадра, и прогнозируемому вперед кодированному изображению из состава ГИ.

Восьмой способ кодирования изображений согласно настоящему изобретению отличается тем, что при использовании четвертого способа кодирования общее количество пригодных данных пропорционально распределяется в зависимости от количества данных в первом потоке бит в каждый отрезок времени, на основе чего скорость кодирования определяется в каждый заданный отрезок времени.

Девятый способ кодирования изображений согласно настоящему изобретению включает операции определения предела кодирования в каждом заранее заданном массиве изображения во входном видеосигнале, установки скорости кодирования в каждом заранее заданном массиве изображения на основе предела кодирования и общего количества пригодных данных и выполнения кодирования входного видеосигнала таким образом, что скорости кодирования соответственных массивов изображения соответствуют установленной скорости кодирования для каждого массива изображения.

Десятый способ кодирования изображений согласно настоящему изобретению отличается тем, что при использовании девятого способа заранее задаваемым массивом изображения является кадр.

Одиннадцатый способ кодирования изображений согласно настоящему изобретению отличается тем, что при осуществлении девятого способа кодирования заранее заданным массивом изображения выбирается ГИ, состоящая из множества кадров.

Двенадцатый способ кодирования изображений согласно настоящему изобретению отличается тем, что при осуществлении девятого способа кодирования изображения выполняется заранее задаваемое прогнозирующее кодирование и/или заранее задаваемое преобразовательное кодирование по меньшей мере части входного видеосигнала для формирования коэффициентных данных и квантования этих данных с фиксированным шагом для определения предела кодирования.

Первое устройство для кодирования изображений согласно настоящему изобретению содержит первое кодирующее средство для кодирования по меньшей мере части входного видеосигнала и последующего генерирования первых кодированных данных, средство управления кодированием для определения скорости кодирования в каждый заранее задаваемый отрезок времени на основе наличия в каждый заранее задаваемый момент количества первых кодированных данных, поступающих от первого кодирующего средства, и общего количества пригодных к использованию данных и вторые кодированные данные для кодирования входного видеосигнала в каждый заранее задаваемый отрезок времени на основе скорости кодирования, заранее задаваемой в каждый отрезок времени средством управления кодированием для генерирования вторых кодированных данных.

Второе устройство кодирования изображения согласно настоящему изобретению отличается тем, что в первом устройстве кодирования изображения первое кодирующее средство содержит средство

RU 2123769 C1

RU 2123769 C1

квантования, служащее для квантования по меньшей мере части входного видеосигнала с фиксированным шагом.

Третье устройство кодирования изображений согласно настоящему изображению отличается тем, что при используемом первом устройстве кодирования изображения средство управления кодированием осуществляет пропорциональное распределение общего количества пригодных данных в зависимости от наличия определенного количества данных в каждый заранее заданный момент (отрезок) времени с тем, чтобы определять скорость кодирования в каждый заранее задаваемый отрезок времени.

Четвертое устройство кодирования изображений согласно настоящему изобретению содержит средство кодирования, служащее для выполнения заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования по меньшей мере части входного видеосигнала для генерирования коэффициентных данных, первое квантующее средство для квантования первых коэффициентных данных, поступающих с первого кодирующего средства, с фиксированным шагом квантования для вырабатывания первых квантованных данных, первое средство неравномерного кодирования (с переменной длиной) для обеспечения возможности для квантованных данных, приходящих с первого средства квантования, подвергнуться неравномерному кодированию (с переменной длиной) для формирования первого потока бит, средство управления кодированием для определения скорости кодирования в каждый заранее задаваемый отрезок времени на основе количества данных в первом потоке бит, поступающем с первого средства неравномерного кодирования (с переменной длиной), и общего количества пригодных для использования данных, второе кодирующее средство, служащее для выполнения заранее задаваемого прогнозирующего кодирования и/или прогнозирующего преобразовательного кодирования входного видеосигнала для выработки вторых коэффициентных данных, второе средство квантования, квантующее вторые коэффициентные данные с выхода второго средства кодирования с шагом квантования, задаваемым на основе скорости кодирования в каждый заранее задаваемый отрезок времени средством управления кодированием для выработки вторых квантованных данных, и второе средство неравномерного кодирования (с переменной длиной), позволяющее вторым квантованным данным с выхода квантующего средства пройти кодирование с переменной длиной для формирования второго потока бит.

Пятое устройство кодирования изображения согласно настоящему изобретению отличается тем, что при использовании четвертого устройства кодирования средство управления кодированием определяет скорость кодирования в каждом кадре на основе количества данных в каждом кадре первого потока бит и общего количества пригодных данных.

Шестое устройство кодирования изображений согласно настоящему изобретению отличается тем, что при использовании четвертого устройства кодирования средство управления кодированием определяет скорость кодирования в каждой ГИ на основе количества данных по крайней мере в части каждой ГИ, состоящей из множества кадров в первом потоке бит, и в общем количестве пригодных данных.

Седьмое устройство кодирования изображений согласно настоящему изобретению отличается тем, что при использовании шестого устройства кодирования изображений средство управления кодированием определяет скорость кодирования в каждой ГИ на основе количества данных в закодированном внутри кадра изображении и прогнозированно закодированном вперед изображении в ГИ.

Восьмое устройство кодирования изображений согласно настоящему изобретению средство управления кодированием пропорционально распределяет общее количество пригодных данных в зависимости от количества данных в первом потоке бит в каждый заранее задаваемый отрезок времени, определяя таким образом скорость кодирования в каждый задаваемый момент времени.

Девятое устройство кодирования изображения согласно настоящему изобретению содержит средство вычисления предела, служащее для расчета (определения) предела кодирования в каждом заранее задаваемом массиве изображения входного видеосигнала, средство установки скорости кодирования, служащее для установки скорости кодирования в каждом заранее задаваемом массиве изображения на основе показателей предела кодирования, поступающих со средства вычисления предела и общего количества пригодных к использованию данных, и кодирующее средство, дающее возможность входному видеосигналу пройти кодирование таким образом, что скорости кодирования соответствующих массивов изображения находятся в соответствии со скоростью кодирования каждого массива изображения, установленной средством установки скорости кодирования.

Десятое устройство кодирования изображения согласно настоящему изобретению отличается тем, что при использовании девятого устройства кодирования изображения средство вычисления помех определяет (предел) кодирования для каждого кадра.

Одиннадцатое устройство для кодирования изображений согласно настоящему изобретению отличается тем, что средство вычисления определяет (предел) кодированию для каждой ГИ, состоящей из множества кадров.

Двенадцатое устройство кодирования изображения согласно настоящему изобретению отличается тем, что при использовании девятого устройства кодирования изображения средство вычисления предела выполняет заранее задаваемое прогнозирующее кодирование и/или заранее задаваемое преобразовательное кодирование по меньшей мере части входного видеосигнала для вырабатывания коэффициентных данных

RU 2123769 C1

RU 2123769 C1

и последующего квантования этих данных при фиксированном шаге квантования, на основе чего вычисляется (определяется) предел кодирования.

Первый носитель информации для регистрации изображения согласно настоящему изобретению отличается тем, что регистрируется (записывается) второй поток бит, получаемый путем кодирования по крайней мере части входного видеосигнала для генерации первых кодированных данных, по которым определяют скорость кодирования в каждый заранее задаваемый отрезок времени на основе наличия в каждый заранее задаваемый отрезок времени количества первых кодированных данных и общего количества пригодных для использования данных с тем, чтобы кодировать входной видеосигнал в каждый заранее задаваемый отрезок времени на основе кодированных данных.

Второй носитель информации для регистрации (записи) изображения согласно настоящему изобретению отличается тем, что регистрируется второй поток бит, получаемых путем выполнения заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования по крайней мере части входного видеосигнала для выработки первых коэффициентных данных и последующего квантования этих данных с фиксированным шагом квантования с тем, чтобы получить первые квантованные данные, которые получат возможность подвергнуться неравномерному кодированию (с переменной длиной) для формирования первого потока бит, по которому определяется скорость кодирования в каждый заранее задаваемый отрезок времени на основе количества данных в первом потоке бит и общего количества пригодных к использованию данных с тем, чтобы затем выполнить заранее задаваемое прогнозирующее кодирование и/или заранее задаваемое преобразовательное кодирование входного видеосигнала для выработки вторых коэффициентных данных и их последующего квантования с фиксированным шагом, задаваемым на основе скорости кодирования в каждый заранее задаваемый отрезок времени, и последующей выработки вторых квантованных данных, которые получат возможность пройти неравномерное кодирование (с переменной длиной).

Третий носитель информации для регистрации (записи) изображения согласно настоящему изобретению отличается тем, что на регистрацию выводятся кодирование данные, полученные при определении предела кодирования в каждом заранее задаваемом массиве изображения во входном видеосигнале с тем, чтобы установить скорость кодирования в каждом заранее задаваемом массиве изображения на основе показателей предела кодирования и количества пригодных к использованию данных и выполнить кодирование входного видеосигнала таким образом, чтобы скорости кодирования соответствующих массивов изображения находились в соответствии с установленной для каждого массива изображения скоростью кодирования.

В соответствии с первым способом кодирования изображения согласно настоящему изобретению скорость кодирования в каждый заранее заданный отрезок времени определяется на основе количества данных в каждый заранее задаваемый отрезок времени из числа первых кодированных данных, получаемых путем кодирования по крайней мере части входного видеосигнала, и общего количества пригодных данных для дальнейшего кодирования входного видеосигнала в каждый заранее задаваемый отрезок времени на основе определенной скорости кодирования и для последующего вырабатывания вторых кодированных данных.

В соответствии с третьим способом кодирования изображений согласно настоящему изобретению при осуществлении первого способа кодирования изображения общее количество пригодных данных пропорционально распределяется в зависимости от количества данных в каждый заранее задаваемый отрезок времени для определения скорости кодирования в каждый заранее заданный отрезок времени для дальнейшего кодирования входного видеосигнала в каждый заданный отрезок времени на основе определенной скорости кодирования и для вырабатывания вторых кодированных данных.

В соответствии с четвертым способом кодирования изображения согласно настоящему изобретению сначала по крайней мере только часть входного сигнала подвергается последовательно заранее задаваемому прогнозирующему кодированию и/или заранее задаваемому преобразовательному кодированию, обработке в виде квантования и неравномерному кодированию (с переменной длиной) при фиксированном шаге квантования для выработки первого потока бит и определения скорости кодирования на основе количества данных в первом потоке бит и общего количества пригодных данных. Затем для формирования второго потока бит весь входной видеосигнал подвергается обработке в виде заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования, квантования и кодированию с переменной длиной при фиксированном шаге квантования, устанавливаемого на основе скорости кодирования в каждый заранее задаваемый отрезок времени.

В соответствии с пятым способом кодирования изображений согласно настоящему изобретению при осуществлении четвертого способа кодирования изображений согласно настоящему изобретению скорость кодирования определяется в каждом кадре на основе количества данных в одном кадре потока бит и общего количества пригодных данных. Затем для формирования второго потока бит входной видеосигнал подвергается заранее задаваемому прогнозирующему кодированию и/или заранее задаваемому преобразовательному кодированию, а также обработке квантованием и неравномерным кодированием (с переменной длиной) при фиксированном шаге квантования, задаваемом на основе скорости кодирования в каждом кадре.

В соответствии с шестым способом кодирования изображений согласно настоящему изобретению берут за основу

RU 2123769 C1

RU 2123769 C1

четвертый способ кодирования изображения и определяют скорость кодирования в каждой ГИ на основе количества данных по меньшей мере в части каждой ГИ, состоящей из множества кадров первого потока бит, и общего количества пригодных данных.

В соответствии с седьмым способом кодирования изображений согласно настоящему изобретению за основу берут шестой способ кодирования изображения и определяют скорость кодирования в каждой ГИ на основе количества данных относительно закодированного внутри кадра изображения и прогнозированного вперед изображения в составе ГИ. Затем входной видеосигнал подвергается обработке в виде (для формирования второго потока бит) заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования, квантизации с фиксированным шагом, задаваемым на основе скорости кодирования в каждой ГИ, а также обработке кодированием с переменной длиной.

В соответствии с восьмым способом кодирования изображений согласно настоящему изобретению за основу берут четвертый способ кодирования изображений и общее количество пригодных к использованию данных пропорционально распределяют в зависимости от количества данных в первом потоке бит в каждый заранее задаваемый отрезок времени для определения скорости кодирования в каждый заранее задаваемый отрезок времени. Затем для формирования второго потока бит входной видеосигнал подвергается обработке в виде заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования, квантизации с фиксированным шагом квантизации, задаваемым на основе скорости кодирования в каждый заранее задаваемый отрезок времени и кодирования с переменной длиной.

В соответствии с девятым способом кодирования изображений согласно настоящему изобретению предел кодирования в каждом заранее задаваемом массиве изображения во входном видеосигнале определяют для установления скорости кодирования в каждом заранее задаваемом массиве изображения на основе показателей предела кодирования и общего количества пригодных к использованию данных. Затем входной видеосигнал кодируется таким образом, что скорости кодирования в соответствующих массивах изображения находятся в соответствии с установленной скоростью кодирования для каждого массива изображения.

В соответствии с десятым способом кодирования изображения согласно настоящему изобретению за основу берут девятый способ кодирования изображений и определяют помехи кодированию для каждого кадра входного видеосигнала для определения скорости кодирования в каждом кадре. Затем входной видеосигнал кодируется так, что скорости кодирования соответствующих кадров соответствуют установленной скорости кодирования в каждом кадре.

В соответствии с одиннадцатым способом кодирования изображений за основу снова берут девятый способ и определяют порог кодирования в каждой ГИ входного видеосигнала для определения скорости кодирования в каждой ГИ. Затем входной видеосигнал кодируется так, что скорости кодирования в Группах Изображений находятся в соответствии с установленной для каждой ГИ скоростью кодирования.

В соответствии с двенадцатым способом кодирования изображений согласно настоящему изобретению также за основу берут девятый способ кодирования изображений, и коэффициентные данные, получаемые путем заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования по меньшей мере части входного видеосигнала, квантуют с фиксированным шагом для определения таким образом предельных показателей. Затем входной видеосигнал кодируется так, что скорости кодирования соответствующих массивов изображений поддерживаются в соответствии со скоростью кодирования каждого массива изображения.

Далее, в соответствии с первым устройством кодирования изображений согласно настоящему изобретению скорость кодирования в каждый заранее задаваемый отрезок времени определяется на основе количества данных в каждый заданный отрезок времени в составе первых кодированных данных, полученных путем кодирования по меньшей мере части входного видеосигнала в каждый заранее задаваемый отрезок времени, и общего количества пригодных данных для дальнейшего кодирования всего входного видеосигнала в каждый заранее задаваемый отрезок времени на основе определенной скорости кодирования и последующей выработки вторых кодированных данных.

В соответствии с вторым устройством кодирования изображений согласно настоящему изобретению за основу берут первое устройство и по меньшей мере часть входного видеосигнала квантуется при фиксированном шаге квантования для выработки таким образом первых кодированных данных, по которым определяют скорость кодирования, с которой кодируется входной видеосигнал в каждый заранее задаваемый отрезок времени, в результате чего вырабатываются вторые кодированные данные.

В соответствии с третьим устройством кодирования изображений согласно настоящему изобретению используется первое устройство кодирования изображения, в котором общее количество пригодных для использования данных пропорционально распределяется в зависимости от количества данных в каждый заранее задаваемый отрезок времени для определения скорости кодирования в каждый заранее задаваемый момент времени и затем с этой скоростью кодируется входной видеосигнал в каждый заданный момент, в результате чего вырабатываются вторые кодированные данные.

В соответствии с четвертым устройством кодирования изображения согласно настоящему изобретению по меньшей мере часть входного видеосигнала подвергается обработке в виде заранее заданного

RU 2123769 C1

RU 2123769 C1

прогнозирующего кодирования и/или заранее заданного преобразовательного кодирования, а также квантования с фиксированным шагом и неравномерного кодирования (с переменной длиной) для формирования первого потока бит с тем, чтобы определить скорость в каждый заранее задаваемый отрезок времени на основе количества данных в первом потоке бит и общего количества данных, пригодных к использованию. Затем заранее задаваемому прогнозируемому кодированию и/или преобразовательному кодированию, а также квантованию и кодированию с переменной длиной с шагом квантования, задаваемым на основе скорости кодирования в каждый задаваемый отрезок времени, подвергается весь входной видеосигнал, в результате чего формируется второй поток бит.

В соответствии с пятым устройством кодирования изображения согласно настоящему изобретению во взятом за основу четвертом устройстве кодирования изображения скорость кодирования определяется в каждом кадре на основе количества данных в каждом кадре первого потока бит и общего количества пригодных данных. Затем обработке в виде заранее задаваемого прогнозирующего кодирования и/или прогнозирующего преобразовательного кодирования, а также квантования с шагом квантования, задаваемым на основе скорости кодирования в каждом кадре, и кодированию с переменной длиной подвергается весь входной видеосигнал, в результате чего формируется второй поток бит.

В соответствии с шестым устройством кодирования изображений согласно настоящему изобретению в принятом за основу четвертом устройстве кодирования изображений скорость кодирования определяется в каждой ГИ на основе количества данных по меньшей мере в части каждой ГИ, состоящей из множества кадров в первом потоке бит, и общего количества пригодных данных. Затем обработке подвергается весь входной видеосигнал, а именно заданному прогнозирующему кодированию и/или заданному преобразовательному кодированию, квантованию с шагом квантования, задаваемым на основе скорости кодирования в каждой ГИ, и кодированию с переменной длиной, в результате чего формируется второй поток бит.

В соответствии с седьмым устройством кодирования изображений согласно настоящему изобретению в принятом за основу шестом устройстве кодирования изображения скорость кодирования в каждой ГИ определяется на основе количества данных в изображениях, кодированных внутри кадра, и прогнозируемых вперед кодированных изображениях в составе ГИ. Затем входной видеосигнал подвергается обработке в виде заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования, квантования с шагом, определяемым скоростью кодирования в каждой ГИ, и кодирования с переменной длиной, в результате чего формируется второй поток данных.

В соответствии с восьмым устройством кодирования изображения согласно настоящему изобретению в принимаемом за основу четвертом устройстве кодирования изображения общее количество пригодных к использованию данных пропорционально распределяется в зависимости от количества данных в первом потоке бит в каждый заранее задаваемый отрезок времени для определения скорости кодирования в каждый задаваемый момент. Затем входной видеосигнал подвергается обработке в виде заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования, квантования с шагом, определяемым на основании скорости кодирования в каждый заранее заданный момент, и неравномерному кодированию (с переменной длиной), в результате чего формируется второй поток данных.

В соответствии с девятым устройством кодирования изображения согласно настоящему изобретению предел кодирования определяется в каждом задаваемом массиве изображения входного видеосигнала для установления скорости кодирования в каждом заданном массиве изображения на основе параметров предела кодирования и общего количества данных, пригодных к использованию. Затем входной видеосигнал кодируется так, что скорости кодирования соответственных массивов изображения поддерживаются в соответствии с установленной скоростью кодирования в каждом массиве изображения.

В соответствии с девятым устройством кодирования изображения согласно настоящему изобретению в принимаемом за основу девятом устройстве кодирования порог кодирования определяется в каждом кадре входного видеосигнала для определения скорости кодирования в каждом кадре. Затем входной видеосигнал кодируется так, что скорости кодирования в соответственных кадрах находятся в соответствии с установленной скоростью кодирования в каждом кадре.

В соответствии с одиннадцатым устройством кодирования изображения согласно настоящему изобретению за основу берут снова девятое устройство кодирования и помехи кодированию определяют в каждой ГИ входного видеосигнала для определения скорости кодирования в каждой GOP. Затем входной видеосигнал кодируется так, что скорости кодирования соответственных ГИ поддерживаются в соответствии со скоростью кодирования, установленной для каждой ГИ.

В соответствии с двенадцатым устройством кодирования изображений согласно настоящему изобретению за основу также берут девятое устройство кодирования, и коэффициентные данные, получаемые путем выполнения заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования по меньшей мере части входного видеосигнала, квантуют с фиксированным шагом для определения предела кодирования. Затем входной видеосигнал кодируется таким образом, что скорости кодирования соответственных массивов изображения находятся в соответствии со скоростью кодирования, установленной для каждого массива изображения.

Кроме того, в соответствии с первым носителем для регистрации (записи)

RU 2123769 C1

RU 2123769 C1

изображения согласно настоящему изобретению на регистрацию выводится второй поток бит, получаемых путем определения скорости кодирования в каждый заранее задаваемый отрезок времени на основе количества данных в каждый момент или отрезок времени из состава первых кодированных данных, получаемых кодированием по меньшей мере части входного видеосигнала, и общего количества пригодных к использованию данных с тем, чтобы кодировать входной видеосигнал в каждый заранее задаваемый момент времени на основе определенной скорости кодирования.

В соответствии с вторым носителем информации для регистрации изображения согласно настоящему изобретению регистрируется второй поток бит, получаемый путем выполнения заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования по меньшей мере части входного видеосигнала, а также квантования и неравномерного кодирования (с переменной длиной) с фиксированным шагом для генерации первого потока бит с целью определения скорости кодирования в каждый заданный отрезок времени на основе количества данных в первом потоке бит и общего количества пригодных данных с тем, чтобы подвергнуть обработке входной видеосигнал, а именно заранее задаваемому прогнозирующему кодированию и/или заранее задаваемому преобразовательному кодированию, а также квантованию и кодированию с переменной длиной с шагом квантования, определяемым на основе скорости кодирования в каждый заранее задаваемый отрезок.

В соответствии с третьим носителем информации для регистрации изображения согласно настоящему изобретению на регистрацию выводятся кодированные данные, получаемые при определении предела кодирования в каждом заданном массиве изображения входного видеосигнала с тем, чтобы установить скорость кодирования каждого заранее задаваемого массива изображения на основе параметров предела кодирования и общего количества данных, пригодных для использования, чтобы кодировать входной видеосигнал таким образом, чтобы скорости кодирования соответственных массивов изображения были в соответствии со скоростью кодирования каждого массива изображения.

Краткое описание фигур.

Фиг. 1 - блок-схема, показывающая связи основных частей устройства кодирования изображений.

Фиг. 2 - блок-схема алгоритма, поясняющего работу первой кодирующей схемы, входящей в состав устройства кодирования по фиг. 1.

Фиг. 3 - блок-схема алгоритма, поясняющего работу второй схемы кодирования, входящей в состав устройства кодирования изображения по фиг. 1.

Фиг. 4 - относительное расположение изображений в составе группы изображений GOP.

Фиг. 5 - относительное расположение изображений и сигнал управления кодирования в каждой группе изображений GOP.

Фиг. 6 - блок-схема алгоритма, также поясняющего работу схемы кодирования, входящей в состав устройства кодирования изображений по фиг. 1.

Фиг. 7 - вид изображения для пояснения принципа прогнозирующего кодирования.

Фиг. 8 - диаграмма изображений, поясняющая принцип прогнозирующего кодирования с компенсацией движения.

Фиг. 9 - блок-схема устройства (блока) кодирования изображений и устройства (блока) декодирования изображений согласно настоящему изобретению.

Фиг. 10 - конфигурация кадра, макро-блока и секции.

Фиг. 11 - блок-схема обычного (известного) кодера.

Фиг. 12 - блок-схема обычного (известного) декодера.

Наилучший вариант реализации настоящего изобретения

Далее приводится описание со ссылкой на прилагаемые чертежи одного из вариантов способа кодирования изображений, устройства кодирования изображений и носителя информации для регистрации (записи) изображений согласно настоящему изобретению.

Устройство кодирования изображений согласно настоящему изобретению содержит, как показано на фиг. 10, например, первую схему кодирования 10 для кодирования входного видеосигнала и выработки первых кодированных данных, схему кодирования 30 изображений для определения скорости кодирования в каждый заранее задаваемый отрезок времени на основе количества кодированных данных в каждый заранее задаваемый момент, поступающих с первой кодирующей схемы 10, и общего количества пригодных к использованию данных и вторую схему кодирования 40 для кодирования входного видеосигнала в каждый заранее задаваемый отрезок времени на основе скорости кодирования, определяемой схемой управления кодированием 30 и последующей выработки вторых кодированных данных.

Более конкретно, первая схема 10 кодирования содержит, как показано на фиг. 1, ЗУ 12 кадровой памяти для хранения входных данных изображений, представляющих собой входной видеосигнал, схему 11 определения вектора движения, служащую для обнаружения этого вектора во входных данных изображений, хранящихся в ЗУ 12 кадровой памяти, ЗУ 22 кадровой памяти, хранящее прогнозированные данные изображений, схему 23 компенсации движения, служащую для выполнения компенсации движения прогнозированных данных изображений, которые были считаны из ЗУ 22 кадровой памяти на основе данных о векторе движения, поступающих со схемы 11 обнаружения вектора движения, схему 14 прогнозирующего кодирования для кодирования входных данных изображения на основе прогнозированных данных изображений с компенсацией движения, поступающих со схемы 23 компенсации движения, схему 15 ДКП, выполняющую кодирование разностей в виде Дискретного Косинусного Преобразования (далее по тексту - ДКП) разностей и т.п., которые называют прогнозируемыми ошибками,

RU 2123769 C1

RU 2123769 C1

поступающих со схемы 14 прогнозирующего кодирования, схему 16 квантования для квантования коэффициентных данных со схемы 15 ДКП, с фиксированным шагом квантования для выработки квантованных данных, схему 17 неравномерного кодирования (с переменной длиной) (далее по тексту - НК), позволяющую квантованным данным со схемы 16 подвергнуться обработке НК с тем, чтобы выдать на выход кодированные с переменной длиной данные, схему 18 инверсного квантования данных, поступающих со схемы 16 квантования для воспроизведения коэффициентных данных, схему 20 Инверсного Дискретного Косинусного Преобразования (далее по тексту ИДКП), служащую для выполнения декодирования, например, в виде ИДКП обработки коэффициентных данных со схемы 18 инверсного квантования, и схему 21 суммирования для суммирования данных с выхода схемы 20 ИДКП и прогнозируемых данных изображения с компенсацией движения с выхода схемы 23 компенсации движения, в результате чего вырабатываются прогнозируемые данные, относящиеся к данным следующего изображения, передаваемых затем в ЗУ кадровой памяти 22.

Далее, вторая схема кодирования 40 содержит, как показано на фиг. 1, элемент задержки 43 для задержки входных данных изображения, ЗУ 52 кадровой памяти для хранения прогнозируемых данных изображения, схему 53 компенсации движения прогнозируемых данных изображения, которые были считаны из ЗУ 52 кадровой памяти с учетом вектора движения со схемы 11 обнаружения этого вектора, схему 44 прогнозирующего кодирования входных данных изображений, задержанных элементом задержки 43, на основе прогнозируемых данных изображений с компенсацией движения, поступающих со схемы 53 компенсации движения, схему 45 ДКП, выполняющую кодирование, например ДКП-обработку разностей и т.п., поступающих со схемы 44 прогнозирующего кодирования, для выработки коэффициентных данных, схему 33 установки шкалы квантования (шага квантования) на основе скорости кодирования, определяемой схемой 30 управления кодированием, схему 46 квантования коэффициентных данных, поступающих со схемы 45 ДКП, с шагом квантования, задаваемым схемой 33 установки шкалы квантования, схему 47 НК, дающую возможность подвергнуть квантованные данные со схемы 46 НК-обработке и вывести после этого данные на выход, буфер передачи 49 для временного хранения данных с выхода схемы 47 НК для последующего вывода их с фиксированной скоростью на передачу, схему 48 для инверсного квантования данных со схемы квантования 46 и дальнейшего воспроизведения коэффициентных данных, схему ИДКП 50, выполняющую декодирование, например, в виде ИДКП обработки коэффициентных данных с выхода схемы 48 инверсного квантования с целью воспроизведения разности, и схему 51 суммирования разности с выхода схемы 50 ИДКП и прогнозируемых данных с компенсацией движения с выхода схемы 53 компенсации движения с тем, чтобы выработать прогнозируемые данные изображений, относящиеся к следующим входным данным изображений, и передать эти прогнозируемые данные в ЗУ 52 кадровой памяти.

В настоящем варианте устройства кодирования обработке, которую выполняет первое средство кодирования 10, а именно прогнозирующему кодированию, ДКП-обработке, квантованию с фиксированным шагом и кодированию (с переменной длиной) подвергается входной видеосигнал, скорость кодирования битов определяется средством управления кодированием 30 на основе количества кодированных с переменной длиной данных, которые представляют собой первый получаемый поток бит, и емкости (по данным) носителя 55 записанного изображения, например оптического диска, магнитного диска или магнитной ленты и т.п., или же общего количества пригодных к использованию данных, определяемого скоростью передачи бит через передающую среду. Соответственно второе средство 40 кодирования выполняет прогнозирующее кодирование, ДКП-обработку, квантование и неравномерное кодирование (с переменной длиной) входных данных изображения во второй раз.

Как показано на фиг. 2, в настоящем варианте реализации изобретения в шаге алгоритма 1 схема квантования 16 из состава первого средства кодирования 10 устанавливает шаг квантования равным, например, 1 для квантования коэффициентных данных, переданных со схемы 15 ДКП для вырабатывания квантованных данных. Счетчик 31 средства управления кодированием 30 подсчитывает количество данных, кодированных с переменной длиной (первый поток данных), получаемых в случае представления возможности кодирования (с переменной длиной) квантованных данных в заранее задаваемый отрезок времени, например за время одного кадра, для определения количества формируемых в каждом кадре количества кодов, указывающих предел (порог) кодирования.

В шаге 2 алгоритма схема 32 вычисления скорости передачи битов определяет количество кодов, предназначенных для каждого кадра на основе предела или порога (количества формируемых кодов) каждого кадра и общего количества пригодных данных.

В шаге 3 алгоритма схема квантования 46, второго средства кодирования 40 квантует данные, приходящие со схемы 45 ДКП, с шагом квантования, задаваемым на основе количества кодов, предназначенных для вырабатывания квантованных данных.

Подаваемые на вход данные изображений временно хранятся в ЗУ 12 кадровой памяти, откуда они затем считываются в соответствии с блочным форматом, что описано в известных технических решениях.

Схема 11 обнаружения вектора движения считывает необходимые данные изображения из ЗУ 12 кадровой памяти массивами макроблоков для обнаружения вектора движения, что было описано выше. Эта схема 11 обнаруживает в массивах макроблоков

RU 2123769 C1

RU 2123769 C1

вектор движения текущего (опорного) изображения, используя расположенное впереди или позади исходное изображение из ЗУ 12 кадровой памяти. Здесь обнаружение вектора движения осуществляется так, что вектор движения, для которого, например, сумма абсолютных значений разностей между кадрами в массивах макроблоков становится минимальной, признается соответствующим искомому вектору движения. Обнаруженный вектор движения передается на схемы 23, 53 и т.п. компенсации движения, а сумма абсолютных значений разностей между кадрами в массивах макроблоков подается на схему 13 прогнозирования внутри кадра/вперед/назад/ в обоих направлениях (схема принятия решения).

Эта схема 13 определяет режим прогнозирования на основе упомянутого выше значения, используя его для управления схемой 14 прогнозирующего кодирования, которая переключает режимы прогнозирования "внутрикадровый/вперед/назад/в обоих направлениях" в массивах блоков.

Прогнозирующая кодирующая схема 14 содержит, как можно видеть из фиг. 1, схемы суммирования 14а, 14б, 14в и селекторный (смены режима) переключатель 14г. Когда работа идет во внутрикадровом режиме, этот переключатель манипулирует непосредственно входными данными изображения, но если же задан режим прогнозирования "вперед/назад/ в обоих направлениях", этот переключатель манипулирует разностями (далее по тексту - разностные данные) в каждом из пикселей входных данных изображения, относящихся к соответственным прогнозируемым изображениям. С селекторного переключателя 14 выбранные им данные подаются на схему 15 ДКП.

Схема 15 ДКП выполняет (в масштабе массивов блоков) ДКП-обработку входных данных изображений или разностных данных, приходящих с селекторного переключателя 14г, с использованием двумерной корреляции видеосигнала, а затем получаемые таким образом коэффициентные данные подаются на схему 16 квантования.

Схема 16 квантования квантует коэффициентные данные, поступающие со схемы 15 ДКП, с шагом квантования, установленным на значение 1 с тем, чтобы передать получаемые таким образом коэффициентные данные на схему НК 17 и схему 18 инверсного квантования.

Схема 17 НК выполняет кодирование с переменной длиной квантованных данных совместно с шагом квантования, параметрами прогнозирования и вектором движения и передает закодированные с переменной длиной данные на средство 30 управления кодированием в качестве первого потока бит.

Средство 30 управления кодированием содержит, как видно из фиг. 1, счетчик 31, служащий для подсчета в каждый задаваемый момент или отрезок времени количества кодированных с переменной длиной данных, поступающих со схемы 17 НК, и вычислительную схему 32, служащую для вычисления (определения) задаваемого (назначаемого) количества кодов на единицу времени на основе количества данных с выхода счетчика 31 и общего количества данных, пригодных к использованию. Счетчик 31 подсчитывает количество данных первого потока бит в каждый заранее задаваемый отрезок времени, например в течение каждого одного кадра, с тем, чтобы вычислять (определять) порог (предел насыщенности) каждого кадра и передать это пороговое значение на схему 32 вычисления скорости следования бит.

Схема 32 вычисляет (определяет) количество кодов, назначаемое для каждого кадра, т.е. среднюю скорость кодирования за время каждого кадра на основе предельной емкости кадра и общего количества пригодных к использованию (используемых) данных, и затем передает значение этого назначаемого количества кодов в схему 33 установки шкалы квантования из состава второго средства кодирования 40.

По существу, схема 32 вычисления скорости кодирования выполняет следующие расчеты. Пусть, например, количество всех кадров есть величина N, общее количество используемых данных есть B, предельная емкость (кодовая емкость) i-го, (i = 0, 1, 2,... N-1) кадра есть d_i и количество назначаемых кодов до i-го кода есть b_i, то когда в виде условия задается, что назначаемая кодовая величина b_i должна быть пропорциональна предельной кодовой емкости d_i, как будет указано в формуле (1), то общее количество данных B может быть вычислено путем сложения задаваемого количества кодов всех кадров, как указано ниже в формуле (2). Величина "a" в следующих формулах есть константа.

$$b_i = a \times d_i \quad ...(1)$$

$$B = \sum_{i=0} bi = \sum_{i=0} axdi = a \quad (2)$$

Соответственно, эта константа "a" может быть вычислена с помощью следующей формулы (3). После подстановки этой константы "a" в формулу (1) назначаемое количество кодов b_i до (относительно) i-го кадра может быть вычислено по формуле (4)

$$a = \frac{B}{\sum_{i=0}^{N-1}} \quad (3)$$

$$bi = di \times \frac{B}{\sum_{i=0}^{N-1}} \quad (4)$$

Таким образом, схема 32 вычисления скорости следования битов увеличивает кодовую величину b_i для кадров изображения сложного вида и уменьшает эту величину для кадров изображения более простого вида.

С другой стороны, схема 18 выполняет инверсное квантование уже квантованных в обычном режиме данных, поступающих со схемы 16 с шагом квантования, который устанавливают равным 1 для воспроизведения коэффициентных данных (при этом добавляются искажения, вызванные квантованием), с выхода схемы 15 ДКП и передачи затем этих данных на схему 20 ИДКП.

Схема 20 ИДКП выполняет обработку ИДКП коэффициентных данных для воспроизведения входных данных изображений, соответствующих выходным данным схемы прогнозирующего кодирования 14 во внутрикадровом режиме, и воспроизводит разностные данные во

RU 2123769 C1

RU 2123769 C1

внутрикадровом/вперед/назад/в обоих направлениях прогнозирующем режимах, передавая затем воспроизведенные данные на схему 21 суммирования.

Если установлен режим прогнозирования "вперед/назад/ в обоих направлениях", на схему 21 суммирования поступают прогнозируемые данные изображений с компенсацией движения со схемы 23 компенсации движения. Схема 21 суммирует эти прогнозированные данные изображения и разностные данные, переданные со схемы 20 ИДКП, в результате чего воспроизводятся данные изображения, соответствующие входным данным изображения.

Воспроизведенные таким образом данные изображения хранятся в ЗУ 22 кадровой памяти. По существу, схема 18 инверсного квантования и суммирующая схема 21 образуют локальную схему декодирования, обрабатывающую квантованные данные с выхода схемы 16 квантования на основе заданного режима прогнозирования с тем, чтобы записать полученные декодированные данные в ЗУ 22 кадровой памяти в качестве вперед или назад прогнозированных изображений. Это ЗУ 22 состоит из множества более мелких ячеек кадровой памяти. Здесь производится переключение банка кадровой памяти. В зависимости от параметров данных, подлежащих кодированию, один очный кадр, например, выдается на выход в качестве прогнозированных вперед или назад данных изображения. Далее, в случае прогнозирования "вперед/назад/ в обоих направлениях" прогнозируемые вперед данные изображения и прогнозируемые назад данные изображения, например, усредняются и эти усредненные данные выдаются на выход. Эти прогнозированные данные изображений представляют собой точно такие же данные, как и данные изображений, воспроизведенные устройством декодирования изображения, которое будет описано ниже. На основе этого прогнозированного изображения следующее подлежащее обработке изображение подвергается кодированию в режиме "вперед/назад/в обоих направлениях".

Ниже описывается работа второго кодирующего средства 40. Здесь следует отметить, что поскольку электронные схемы, за исключением схемы 33 установления шкалы квантования, элемента задержки 43, схемы квантования 46 и буфера передачи 49, образующие второе средство кодирования 40, выполняют те же самые операции, что и аналогичные схемы описанного ранее первого средства кодирования 10, то их описание опущено.

Элемент задержки 43 задерживает входные данные изображения, например, до того момента, когда сигнал управления кодированием пройдет на выход средства управления кодированием 30. Затем в схеме прогнозирующего кодирования 44 и схеме 45 ДКП задержанные входные данные изображения подвергаются обработке в виде прогнозируемого кодирования и ДКП, которые выполняются в соответствии в режимом прогнозирования, задаваемым схемой 13, работающей в режимах прогнозирования "внутрикадровом/вперед/назад/в обоих направлениях", и выполняющей функцию принятия решения. Таким образом вырабатываются коэффициентные данные.

Схема 33 установки шкалы квантования определяет (вычисляет) количество кодов, назначаемых для каждого макроблока (например, величину, получаемую путем деления количества кодов, назначаемых для каждого кадра, на число макроблоков в одном кадре) из переданного количества кодов для каждого кадра для выполнения сравнения между количеством кодов, сформированных в макроблоке, который обнаруживается в обратной связи буфера передачи 49, и количеством кодов, назначаемых для каждого макроблока. Работа схемы 33 установки шкалы квантования состоит в том, что она таким образом задает скорости следования битов при кодировании соответствующих кадров, чтобы они были как можно ближе к средним значениям скорости следования битов при кодировании за время каждого кадра. В том случае, когда количество кодов, сформированных в соответствующем макроблоке оказывается больше, чем количество назначенных (заданных) кодов для каждого макроблока, схема 33 устанавливает шаг квантования для следующего макроблока на большую величину для того, чтобы уменьшить количество кодов, которые будут сформированы в следующем макроблоке, и напротив, если количество кодов, сформированных в соответствующем макроблоке, оказывается меньше, чем количество кодов, заданное для каждого макроблока, эта схема 33 устанавливает уменьшенную величину шага квантования для следующего макроблока для увеличения числа формируемых кодов. Здесь следует отметить, еще одну функцию схемы 33, которая состоит в том, что когда в цепи обратной связи буфера передачи 49 появляются указания о том, что буфер 49 находится в состоянии, близком к переполнению, то эта схема 33 также увеличивает шаг квантования для предотвращения переполнения независимо от результатов сравнения заданного и формируемого фактически количества кодов. С другой стороны, если контур обратной связи буфера передачи 49 сигнализирует о том, что содержимое буфера близко к нижнему пределу, схема 33 установления шкалы квантования уменьшает величину шага квантования для увеличения числа формируемых кодов также независимо от результатов сравнения заданного и фактически формируемого количества кодов. Хотя выше было сказано, что сравнение количества заданных (назначаемых) кодов и фактически формируемых выполняется в каждом макроблоке для переключения величины шага квантования для каждого такого макроблока, подобное же переключение может быть выполнено и для каждой секции. Кроме того, хотя выше было указано, что количество фактически формируемых кодов определяют из содержимого буфера передачи 40, эти же показатели можно получить непосредственно с выхода схемы 47 неравномерного кодирования (с переменной длиной). Схема 33 установления шкалы квантования именно таким образом передает величину шага квантования в схему 46 квантования.

Эта схема квантования 46 квантует

RU 2123769 C1

RU 2123769 C1

коэффициентные данные, получаемые от схемы 45 ДКП, с шагом квантования, величина которого передается со схемы 33 установки шкалы для вырабатывания квантованных данных.

Схема НК 47 дает возможность подвергнуть квантованные данные, приходящие со схемы 46 квантования, неравномерному кодированию (с переменной длиной) с шагом квантования, величина которого передается со схемы 33 установки шкалы квантования, и в режиме прогнозирования "внутрикадрового/вперед/назад/в обоих направлениях", задаваемого схемой 13 прогнозируемого решения, а также с учетом вектора движения со схемы его обнаружения 11, после чего полученные таким образом данные кодирования с переменной длиной пересылаются в буфер 49 передачи как второй поток бит.

В настоящем варианте устройства кодирования изображений, как показано на фиг. 3, после ввода данных изображений через элемент задержки 43 в шаге 1 алгоритма, схема 33 установки шкалы получает от средства 30 управления кодированием значение количества кодов, задаваемых для кадра, подлежащего текущему кодированию в шаге 2 алгоритма, после чего обработка переходит к шагу 3.

В шаге 3 алгоритма схема 44 прогнозируемого кодирования совместно со схемой 47 НК выполняют обработку данных изображения в виде прогнозируемого кодирования и ДКП обработки, а затем квантуют коэффициентные данные с шагом квантования, задаваемым на основе количества задаваемых кодов, что дает возможность выполнить кодирование с переменной длиной. Затем обработка переходит к шагу 4.

В шаге 4 алгоритма, независимо от того, завершено или нет кодирование всех кадров (последовательности их), которые имеют один и тот же размер изображения или обрабатываются при одинаковой скорости передачи бит, принимается решение об окончании алгоритма. Если обработка все же на завершена, алгоритм возвращается к шагу 1. Таким образом реализуется кодирование с переменной скоростью, когда скорость кодирования изменяется в рамках массивов кадров. Соответственно, даже если изображения (кадры) сложного вида поступают в виде последовательности, невозможно обеспечить столь же большую величину шага квантования, как в обычных (известных) устройствах. Так обеспечивается однородное и высокое качество изображения во всех возможных ситуациях.

Буфер передачи 49 временно хранит данные кодирования с переменной длиной, чтобы затем выдать их на считывание с фиксированной скоростью следования бит, обеспечивая упорядоченный вывод кодированных с переменной длиной данных в виде потока бит. Поток бит с выхода буфера передачи 49 мультиплексируется, например, с кодированным аудиосигналом, сигналом синхронизации и т. п. Кроме того, сюда же добавляется код коррекции ошибок и модуляция с заранее заданными параметрами, пригодными для передачи или регистрации данных. Затем обработанный таким образом поток бит передается на устройство декодирования изображения либо через среду распространения, либо регистрируется на носителе записи изображения 55, представляющем собой оптический диск, магнитный диск или магнитную ленту, что уже упоминалось выше и показано на фиг. 1. Поскольку во втором средстве кодирования 40 производится такая обработка в виде кодирования с переменной скоростью, когда количество задаваемых кодов увеличивается при обработке изображений сложного вида и уменьшается при обработке изображений простого вида, то нет необходимости поддерживать высокую и фиксированную скорость во всех случаях во избежание значительного ухудшения качества изображений сложного вида, как это делается в обычных (известных) устройствах. Благодаря этому время регистрации изображения на носителе 55 может быть продлено.

С другой стороны, схема 48 выполняет инверсное квантование данных, квантованных обычным методом и переданных со схемы 46 квантования, причем величина шага квантования при этом та же, что и в схеме 46, что необходимо для воспроизведения коэффициентных данных (при этом добавляются искажения из-за квантования), соответствующих выходным данным схемы 45 ДКП, после чего эти коэффициентные данные пересылаются в схему 50 ИДКП. Кроме того, схема 48 инверсного квантования и суммирующая схема 51 образуют локальную схему декодирования, декодирующую квантованные данные, поступающие с выхода схемы квантования 46, после чего полученное таким образом декодированное изображение записывается в ЗУ 52 кадровой памяти как прогнозированное вперед или назад изображение. Данные изображения, хранящиеся в ЗУ 52, используются в качестве прогнозированного изображения для следующего подлежащего обработке изображения.

Следует отметить, что хотя в вышеупомянутом варианте реализации изобретения задаваемое количество кодов для заранее задаваемого отрезка времени, т. е. средняя скорость кодирования на протяжении заранее заданного отрезка времени, определяется для каждого кадра, причем за заданный отрезок времени принимается длительность кадра, оно (изобретение) не ограничивается только таким вариантом реализации. Например, в качестве заранее задаваемого отрезка времени может быть использована длительность GOP (Группы Изображений) в нормативах так называемой Экспертной Группы по Подвижным Изображениям (ЭГПИ). Название этой группы ЭГПИ (MPEG) используется и как наименование системы кодирования подвижных изображений, исследованной в Рабочей Группе (РГ) 11 Подкомитета 29 (ПК) Объединенного Технического Комитета (ОТК) в составе Международной Организации по Стандартизации (МОС) и Международного Электротехнического Комитета (МЭТК).

Согласно нормативам (ЭГПИ) группа изобаржений GOP содержит по меньшей мере одно так называемое I-изображение и множество P-изображений или B-изображений

RU 2123769 C1

RU 2123769 C1

(не I-изображения). Рассмотрим практический пример, когда GOP состоит из одиночного I-изображения, четырех P-изображений с периодом следования в 3 изображения и десять B-изображений, а схема управления кодированием 30 определяет количество задаваемых кодов для каждой GOP. Здесь I-изображение подвергается внутриполевому или внутрикадровому кодированию. P-изображение есть изображение, прогнозируемое только в направлении вперед и подлежащее внутриполевому или внутрикадровому кодированию.
B-изображение есть изображение, которое может прогнозироваться в направлениях вперед, назад и в обоих направлениях, оно подлежит внутриполевому или внутрикадровому кодированию.

Если далее предполагается, например, как показано на фиг. 5, что два последовательных произвольных изображения в составе ГИ, состоящем из определенного количества изображений, есть повторяющиеся периодически I-изображение и P-изображение, и величина шага квантования установлена равной, например, 1, то первое средство кодирования 10 может выполнить прогнозирующее кодирование, ДКП обработку и кодирование с переменной длиной данных этих I-изображения и P-изображения для выработки кодированных с переменной длиной данных, которые затем будут переданы на средство 30 управления кодированием. Два эти изображения именно типа I и P выбраны с целью проверки сложности рисунка изображения и степени корреляции между кадрами. По количеству формируемых кодов I-изображения возможно определить (оценить) сложность изображения. По количеству формируемых кодов P-изображения можно определить степень корреляции между кадрами. Поскольку последовательность множества кадров имеет в общем-то аналогичные пиктографические элементы, то даже по двум выбранным изображениям можно оценить тенденцию структуры GOP.

Средство 30 управления кодированием подсчитывает (вычисляет) количество данных I-изображений в каждой ГИ, обозначаемое как I_j бит, и количество данных P-изображений P_j бит, и определяет для каждой GOP фактический предел насыщенности (порог) (количество формируемых кодов d_j, где j = 0, 1, 2,...) на основе величин I_j и P_j, а также количество N P-изображений, составляющих GOP, как указано ниже в формуле (5).

$$\text{GOP } d_j = \text{бит } J_i + N^x \text{ бит } P_j \ldots (5)$$

Средство 30 управления кодированием определяет количество кодов, отведенное для каждой, на основе величины порога каждой GOP и общего количества используемых данных, а затем пересылает это значение количества задаваемых кодов к второму средству кодирования 40.

Если принять, что число групп изображений GOP есть M, общее количество пригодных к использованию данных есть B, задаваемое количество кодов применительно к j-му GOP есть GOP b_j, а также задать, что эта величина GOP b_j пропорциональна пороговому значению, как следует из приведенной ниже формулы (6), то общее количество данных B определяется путем суммирования величин b_i все, как записано в формуле (7). Величина a в формуле (6) есть константа.

$$\text{GOP } b_j = a \text{ GOP } d_j \ldots (6)$$

$$B = \sum_{j=0}^{M-1} \text{GOP } b_i = \sum_{j=0}^{M-1} a \text{GOP } d_j = a \sum_{j=0}^{M-1} \text{GOP } d_j \quad (7)$$

Эта константа a может быть определена из формулы (8). Подставляя эту константу в формулу (6), можно по формуле (9) определить задаваемое количество кодов ГИ b_j для j-го ГИ.

$$a = \frac{B}{\sum_{j=0}^{M-1} \text{GOP } d_j} \quad (8)$$

$$\text{GOP } b_j = \text{GOP } d_j \times \left(\frac{B}{\sum_{j=0}^{M-1} \text{GOP } d_j} \right) \quad (9)$$

Итак, средство 30 управления кодированием увеличивает величину количества задаваемых кодов GOP b_j в случае, например, группы изображений, в которых содержатся изображения со сложным рисунком или которые обладают низкой степенью корреляции между кадрами, и напротив, в группе изображений, включающих изображения с простым рисунком или имеющих высокую степень корреляции между кадрами, это средство 30 уменьшает предназначаемое для нее количество кодов.

Когда на второе средство кодирования 40 в шаге 1 алгоритма, приведенного на фиг. 6, через элемент задержки 43 поступают данные изображений, это средство 40 определяет, являются или нет эти данные начальными данными (шаг 2) группы изображений. Если это так, алгоритм обработки переходит к шагу 3. В противном случае алгоритм переходит к шагу 4.

В шаге 3 второе средство кодирования 40 получает от средства управления 30 значение числа кодов, предназначаемых для GOP, подвергаемой текущему кодированию. Затем алгоритм обработки переходит к шагу 4.

В шаге 4 второе кодирующее средство 40 выполняет прогнозирующее кодирование и ДКП-обработку, а также квантует коэффициентные данные с шагом, заданным на основе назначенного числа кодов, что дает возможность выполнить кодирование с переменной длиной. Затем алгоритм обработки переходит к шагу 5.

Здесь схема 33 установки шкалы квантования задает количество кодов для каждого кадра из числа переданного ему общего количества кодов, назначенных для каждой ГИ, принимая во внимание тип изображения (I-изображение, P-изображение, B-изображение) при конкретном кодировании, например тип изображения, показанный на фиг. 4. В практическом аспекте можно сказать, что количество кодов, назначаемых для I-изображений, увеличивается, количество кодов для B-изображений уменьшается, а количество кодов для P-изображений имеет промежуточное между ними значение.

В последующем шаге 5 независимо от того, завершено или нет кодирование всех тех кадров (их последовательности), для которых был задан одинаковый размер изображения

RU 2123769 C1

RU 2123769 C1

или одинаковая скорость передачи бит, принимается соответствующее решение. Если последовательность закончилась, то и обработка считается завершенной. В противном случае алгоритм возвращается к шагу 1. Так реализуется неравномерное кодирование (с переменной длиной), когда скорость кодирования изменяется в каждой GOP. Если даже имеет место последовательность изображений (кадров) со сложным рисунком, то все же нет возможности сделать шаг квантования достаточно большим для таких изображений, как в обычных (известных) устройствах. В результате появляется возможность формирования однородного изображения высокого качества в целом для всех ситуаций. Кроме того, поскольку задаваемое количество кодов для каждой GOP определяется в настоящем варианте устройства по двум изображениям, то по сравнению с вышеописанным вариантом может быть установлена более высокая скорость обработки. Осталось только отметить тот очевидный факт, что количество кодов соответственных групп изображений определяются на основе количества данных всех изображений в составе GOP.

Необходимо отметить, что настоящее изобретение не ограничивается описанными выше вариантами его. Хотя, например, в описанных выше вариантах для преобразовательного кодирования используется метод ДКП, преобразование Страто, преобразование Хаара или преобразование Вавелета также могут быть применены в описанных случаях.

Промышленное применение.

Как должно быть понятно из приведенного выше описания, в настоящем изобретении используется электронная схема и последовательность операций, позволяющие кодировать входной видеосигнал путем выполнения, например, прогнозирующего кодирования, ДКП-обработки и квантования данных с фиксированным шагом квантования, кодирования (с переменной длиной) для выработки первых кодированных данных и последующего определения (вычисления) количества кодов, отводимых (задаваемых) для каждого кадра или каждой GOP, на основе учета количества данных в каждый заранее заданный отрезок времени, например за время длительности каждого кадра или каждой ГИ из первых кодированных данных, и общего количества данных, пригодных для кодирования входного видеосигнала в каждый заранее задаваемый момент на основе назначаемого количества кодов для выработывания вторых кодированных данных. Таким образом осуществляется кодирование с переменной скоростью, когда скорость кодирования изменяется в заранее задаваемый отрезок времени. В результате, если даже имеет место последовательность изображений (кадров) сложного вида, нет возможности обеспечить такое большое значение шага квантования для этих изображений, как в обычных устройствах. Благодаря этому можно достигнуть однородного изображения высокого качества в целом в различных случаях.

Кроме того, поскольку вторые кодированные данные, формирование которых описано выше, имеют переменную скорость, то в случае, когда такие данные регистрируются на носителе записанного изображения, ограниченный объем памяти может быть использован более эффективно, а время регистрации изображения на носителе может быть продлено. Наконец, данные изображения, имеющего высокое качество и однородность во всех возможных случаях, могут быть воспроизведены с носителя записанного изображения.

Формула изобретения:

1. Способ кодирования изображений, включающий операции кодирования по меньшей мере части входного видеосигнала для выработки первых кодированных данных, определения скорости кодирования в каждый заранее задаваемый момент или отрезок времени на основе количества первых кодированных данных и общего количества пригодных к использованию данных, отличающийся тем, что предусматривает кодирование входного видеосигнала в каждый заранее задаваемый отрезок времени на основе скорости кодирования для выработки вторых кодированных данных.

2. Способ по п.1, отличающийся тем, что по меньшей мере часть входного видеосигнала квантуют с фиксированным шагом квантования для выработки первых кодированных данных.

3. Способ по п.1, отличающийся тем, что общее количество пригодных к использованию данных пропорционально распределяют в соответствии с требуемым количеством данных в каждый заранее задаваемый отрезок времени для определения таким образом скорости кодирования в каждый заранее задаваемый отрезок времени.

4. Способ кодирования изображений, включающий операции заранее задаваемого прогнозирующего кодирования или заранее задаваемого преобразовательного кодирования по меньшей мере части входного видеосигнала для вырабатывания первых коэффициентных данных, квантования первых коэффициентных данных с фиксированным шагом квантования для вырабатывания первых квантованных данных, обеспечивая возможности обработки первых квантованных данных в виде неравномерного кодирования (с переменной длиной) для формирования первого потока бит, определения скорости кодирования в каждый заранее задаваемый отрезок времени на основе количества данных в первом потоке бит и общего количества пригодных к использованию данных, отличающийся тем, что прогнозирующее кодирование и/или заранее задаваемое преобразовательное кодирование входного видеосигнала для генерации вторых коэффициентных данных, квантование вторых коэффициентных данных с шагом квантования, задаваемым на основе скорости кодирования в каждый заранее задаваемый отрезок времени для выработки вторых квантованных данных, и обеспечение возможности кодирования с переменной длиной вторых квантованных данных для формирования второго потока бит.

5. Способ по п.4, отличающийся тем, что скорость кодирования определяют в каждом кадре на основе количества данных в каждом кадре первого потока бит и общего количества пригодных к использованию

RU 2123769 C1

RU 2123769 C1

данных.

6. Способ по п.4, отличающийся тем, что скорость кодирования определяют в каждой группе изображений (ГИ) на основе по крайней мере количества части данных каждой группы изображений (ГИ), состоящей из множества кадров из первого потока бит, и на основе общего количества пригодных к использованию данных.

7. Способ по п.6, отличающийся тем, что скорость кодирования в каждой группе изображений (ГИ) определяют на основе количества данных в изображении с внутрикадровым кодированием и изображения, кодированного с прогнозированием вперед в составе группы изображений (ГИ).

8. Способ по п. 4, отличающийся тем, что общее количество пригодных к использованию данных распределяют пропорционально в зависимости от количества данных в первом потоке бит в каждый заранее задаваемый отрезок времени для определения скорости кодирования в каждый заранее задаваемый отрезок времени.

9. Способ кодирования изображений, включающий операции определения предела кодирования в каждом блоке изображения во входном видеосигнале, установки скорости кодирования в каждом заранее задаваемом блоке изображения на основе предела кодирования и общего количества данных, пригодных к использованию, отличающийся тем, что предусматривает кодирование входного видеосигнала таким образом, что скорости кодирования соответственных блоков изображения поддерживают в соответствии с установленной для каждого блока изображения скоростью кодирования.

10. Способ по п.9, отличающийся тем, что в качестве заранее задаваемого блока изображения задают кадр.

11. Способ по п.9, отличающийся тем, что в качестве заранее задаваемого блока изображения задают группу изображений (ГИ), состоящую из множества кадров.

12. Способ по п.9, отличающийся тем, что заранее задаваемому прогнозирующему кодированию и/или заранее задаваемому преобразовательному кодированию подвергают по крайней мере часть входного видеосигнала для выработки коэффициентных данных с тем, чтобы затем произвести квантование этих коэффициентных данных с фиксированным шагом квантования и определить таким образом предел кодирования.

13. Устройство кодирования изображений, содержащее первое средство кодирования по крайней мере части входного видеосигнала для вырабатывания первых кодированных данных, средство управления кодированием, служащее для определения скорости кодирования в заранее задаваемый отрезок времени на основе количества первых кодированных данных в каждый заранее задаваемый отрезок времени, поступающих от первого средства кодирования, и общего количества данных, пригодных для использования, отличающееся тем, что содержит второе средство кодирования входного видеосигнала в каждый заранее задаваемый отрезок времени на основе скорости кодирования в каждый заранее задаваемый отрезок времени, задаваемой средством управления кодированием, для выработки вторых кодированных данных.

14. Устройство по п.13, отличающееся тем, что первое средство кодирования содержит средство для квантования по крайней мере части входного видеосигнала с фиксированным шагом квантования.

15. Устройство по п.13, отличающееся тем, что средство управления кодированием выполняют функцию пропорционального распределения общего количества данных в зависимости от наличия данных в каждый заранее задаваемый отрезок времени, определяя таким образом скорость кодирования в каждый заранее задаваемый отрезок времени.

16. Устройство кодирования изображений, содержащее первое средство для выполнения заранее задаваемого прогнозирующего кодирования и/иди заранее задаваемого преобразовательного кодирования по меньшей мере части входного видеосигнала для выработки первых коэффициентных данных, первое средство квантования первых коэффициентных данных с фиксированным шагом квантования для образования первых квантованных данных, первое средство неравномерного кодирования, обеспечивающее такую обработку квантованных данных, приходящих с первого средства квантования, для формирования первого потока бит, средство управления кодированием для определения скорости кодирования в каждый заранее задаваемый отрезок времени на основе количества данных в первом потоке бит, поступающем с первого средства неравномерного кодирования, и общего количества используемых данных, отличающееся тем, что содержит второе средство кодирования для выполнения заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого прогнозирующего преобразовательного кодирования входного видеосигнала для выработки вторых коэффициентных данных, второе средство квантования для квантования вторых коэффициентных данных, поступающих со второго кодирующего средства, с шагом квантования, задаваемым с учетом скорости кодирования средством управления кодированием в каждый заранее задаваемый отрезок времени для вырабатывания вторых квантованных данных, и второе средство неравномерного кодирования, обеспечивающее такую обработку вторых квантованных данных, поступающих со второго средства квантования для формирования второго потока бит.

17. Устройство по п.16, отличающееся тем, что средство управления кодированием определяет скорость кодирования в каждом кадре на основе количества данных в каждом кадре первого потока бит и общего количества пригодных данных.

18. Устройство по п.16, отличающееся тем, что средство управления кодированием определяет скорость кодирования в каждой группе изображений (ГИ) на основе количества данных по меньшей мере в части каждой группы изображений (ГИ), состоящей из множества кадров первого потока бит, а также общего количества пригодных к

RU 2123769 C1

RU 2123769 C1

использованию данных.

19. Устройство по п.18, отличающееся тем, что средство управления кодированием определяет скорость кодирования в каждой группе изображений (ГИ) на основе количества данных в изображениях внутрикадрового кодирования и в изображениях, кодированных с прогнозированием вперед в составе группы изображений (ГИ).

20. Устройство по п.16, отличающееся тем, что средство управления кодированием пропорционально распределяет общее количество пригодных к использованию данных в зависимости от количества данных в первом потоке бит в каждый заранее задаваемый отрезок времени, определяя таким образом скорость кодирования в каждый заранее задаваемый отрезок времени.

21. Устройство кодирования изображений, содержащее средство вычисления (определения) предела кодирования в каждом заранее задаваемом блоке изображений во входном видеосигнале, средство установки скорости кодирования в каждом заранее задаваемом блоке изображений на основе предельных параметров кодирования, получаемых от средства вычисления этих предельных параметров, с учетом общего количества пригодных к использованию данных, отличающееся тем, что содержит средство кодирования, обеспечивающее возможность такого кодирования входного видеосигнала, когда скорости кодирования соответственных блоков изображения соответствуют скорости кодирования каждого блока изображений, установленной средством установки скорости кодирования.

22. Устройство по п.21, отличающееся тем, что средство вычисления предела кодирования определяет эти параметры кодирования для каждого кадра.

23. Устройство по п.21, отличающееся тем, что средство вычисления предельных параметров кодирования определяет эти параметры для каждой группы изображений (ГИ), состоящей из множества кадров.

24. Устройство по п.21, отличающееся тем, что средство вычисления предела кодирования выполняет заранее задаваемое прогнозирующее кодирование и/или заранее задаваемое преобразовательное кодирование по крайней мере части входного видеосигнала для выработки коэффициентных данных и их последующего квантования с фиксированным шагом, в результате чего вычисляют (определяют) предел кодирования.

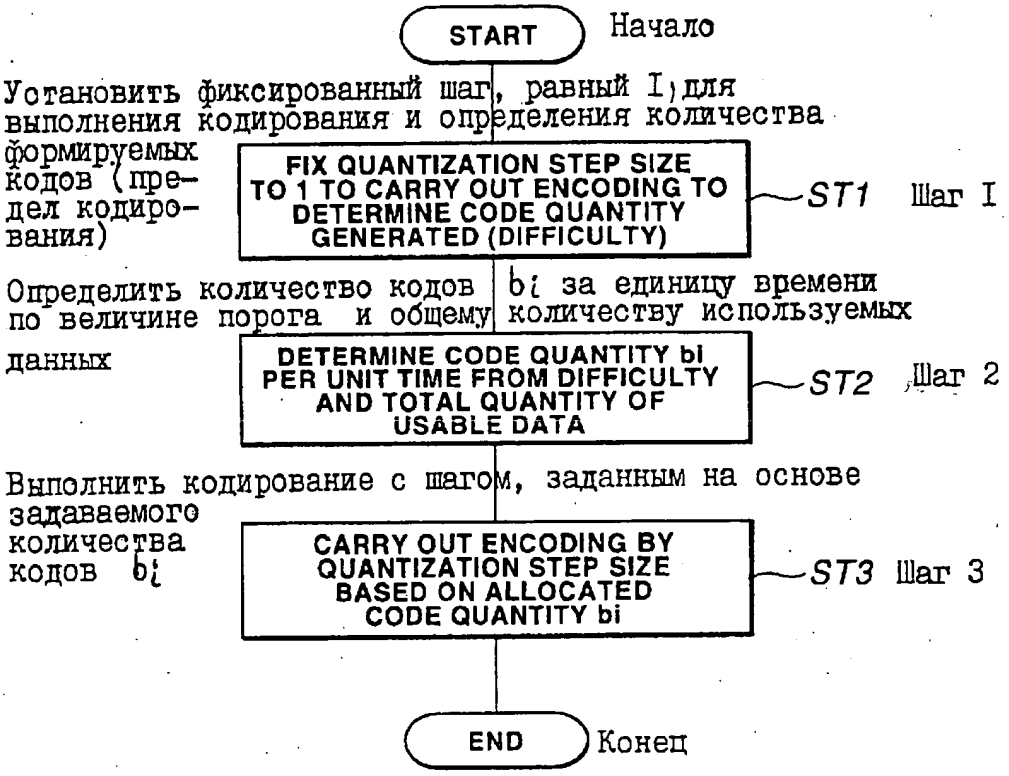
25. Носитель информации для записи изображений, на котором записан второй поток бит, полученный путем кодирования по крайней мере части входного видеосигнала для выработки первых кодированных данных, определения скорости кодирования в каждый заранее задаваемый отрезок времени на основе количества первых кодированных данных в каждый заранее задаваемый отрезок времени, а также общего количества используемых данных, отличающийся тем, что второй поток бит получен в результате кодирования входного видеосигнала в каждый заранее задаваемый отрезок времени на основе скорости кодирования.

26. Носитель информации для записи изображений, на котором записан второй поток бит, полученный путем выполнения заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования по меньшей мере части входного видеосигнала, выработки первых коэффициентных данных и их квантования с фиксированных шагом для выработки первых квантованных данных, обеспечивания возможности неравномерного кодирования первых квантованных данных для формирования первого потока бит, определения скорости кодирования в каждый заранее задаваемый отрезок времени на основе количества данных в первом потоке данных и общего количества пригодных к использованию данных, отличающийся тем, что второй поток бит получен в результате выполнения заранее задаваемого прогнозирующего кодирования и/или заранее задаваемого преобразовательного кодирования входного видеосигнала для выработки вторых коэффициентных данных, квантования вторых коэффициентных данных с шагом квантования, задаваемым на основе скорости кодирования в каждый заранее задаваемый отрезок времени для выработки вторых квантованных данных и обеспечения возможности неравномерного кодирования этих вторых квантованных данных.

27. Носитель информации для записи изображений, на котором записаны кодированные данные, полученные путем определения предела кодирования в каждом заранее задаваемом блоке изображения входного видеосигнала, установления скорости кодирования в каждом заранее задаваемом блоке изображения, на основе величины предела кодирования и общего количества пригодных к использованию данных, отличающийся тем, что записанные данные получены в результате кодирования входного видеосигнала таким образом, что скорости кодирования в каждом из соответственных блоков изображения соответствуют скорости кодирования, установленной для каждого блока изображения.

RU 2123769 C1

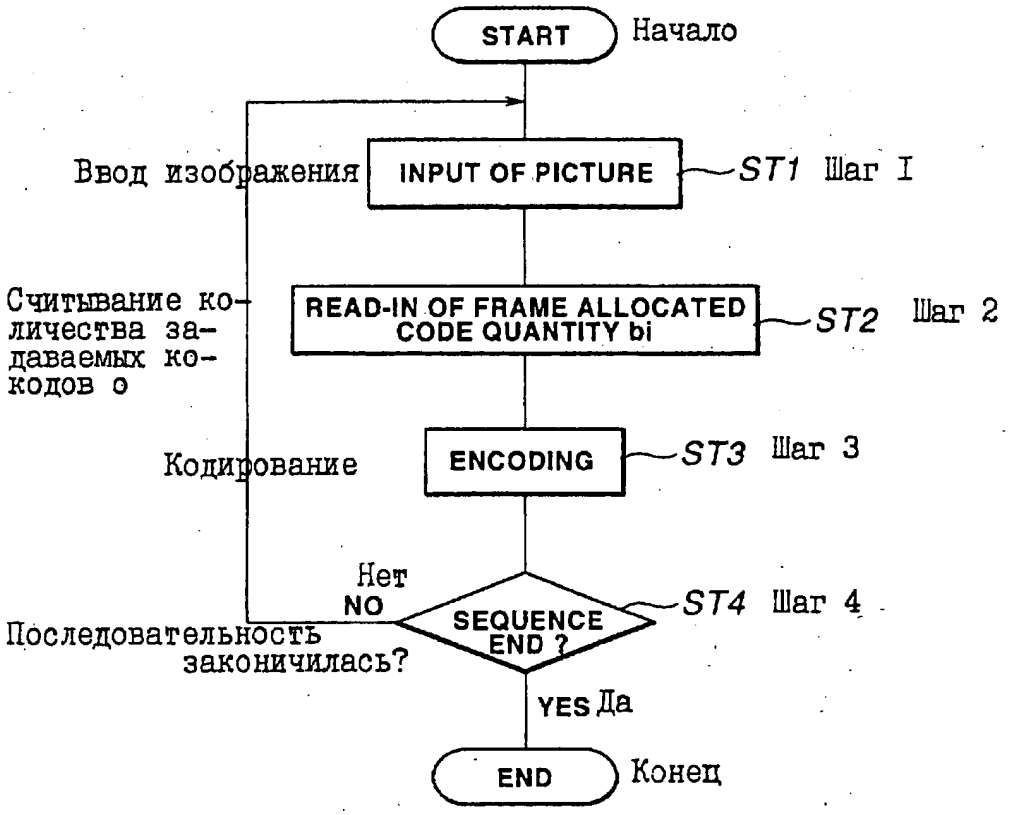
RU 2123769 C1



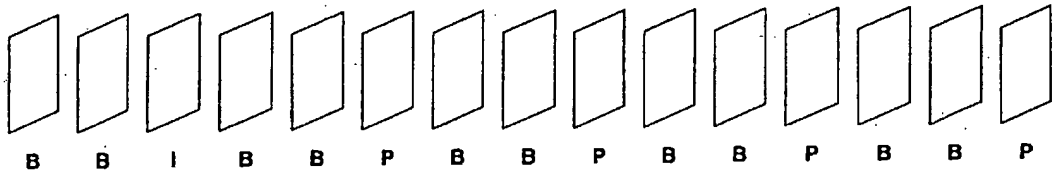
Фиг.2

RU 2123769 C1

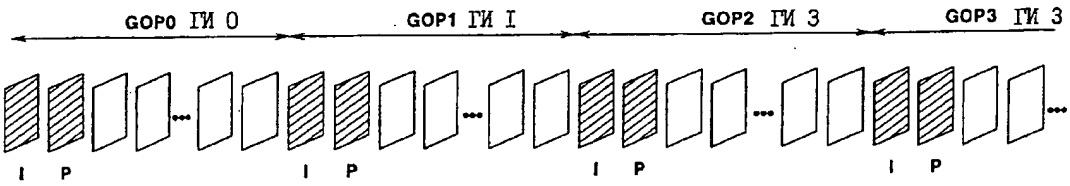
RU 2123769 C1



Фиг.3



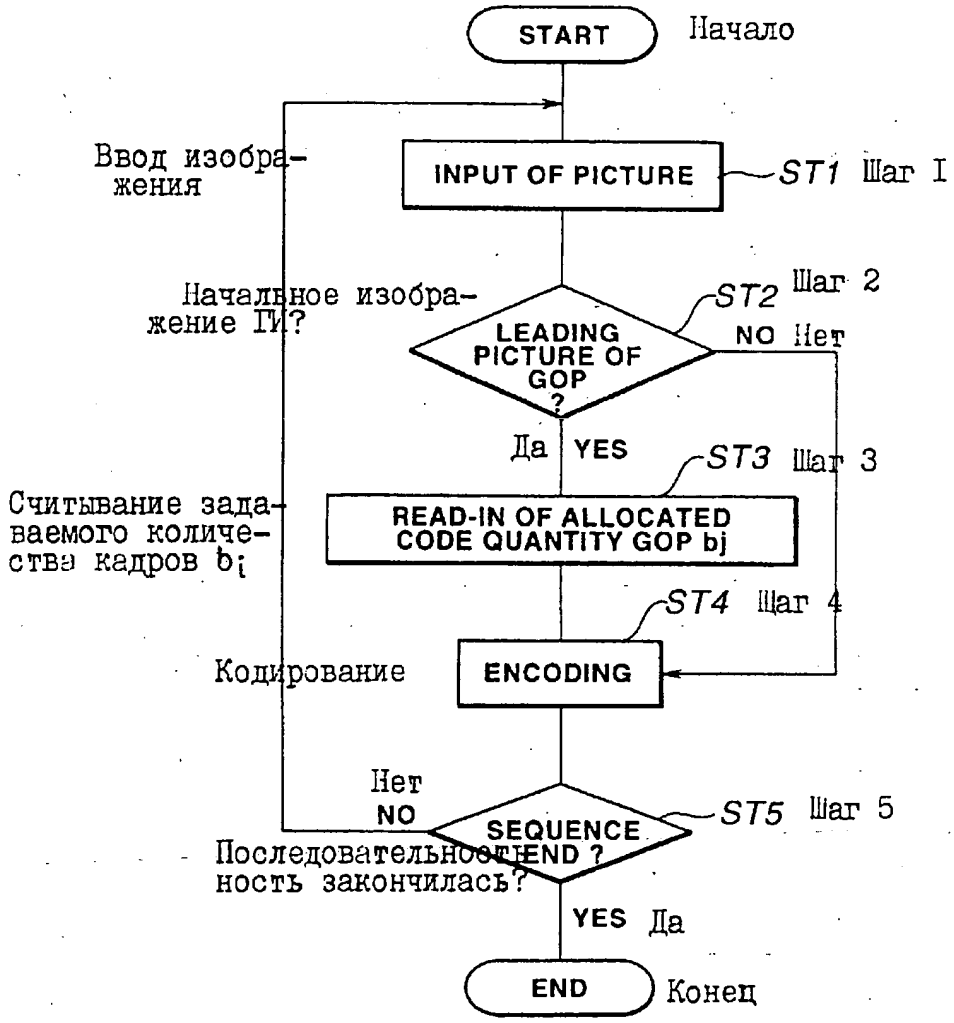
Фиг.4



Фиг.5

RU 2123769 C1

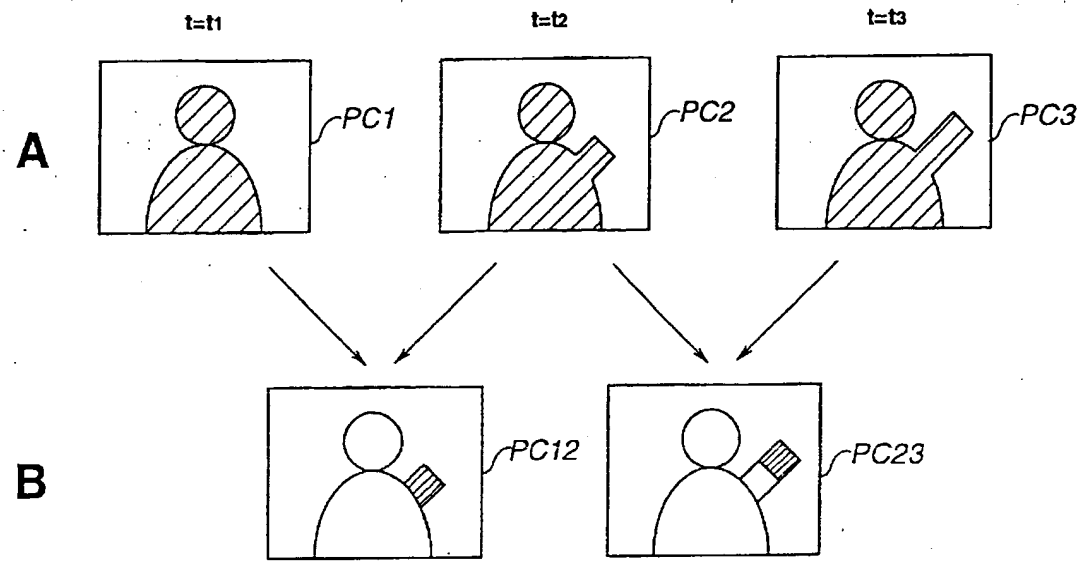
RU 2123769 C1



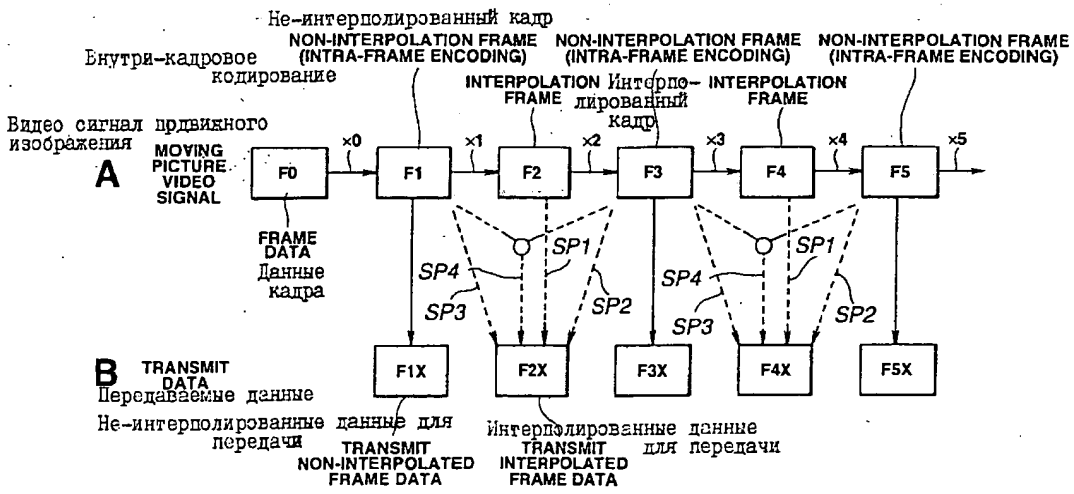
Фиг.6

RU 2123769 C1

RU 2123769 C1



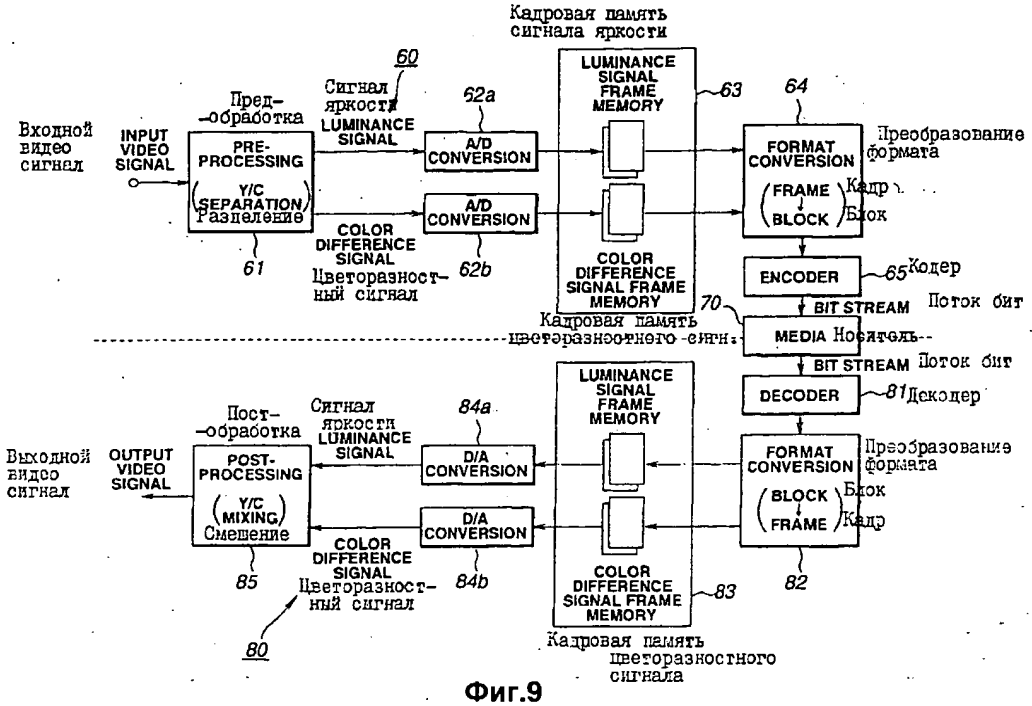
Фиг.7



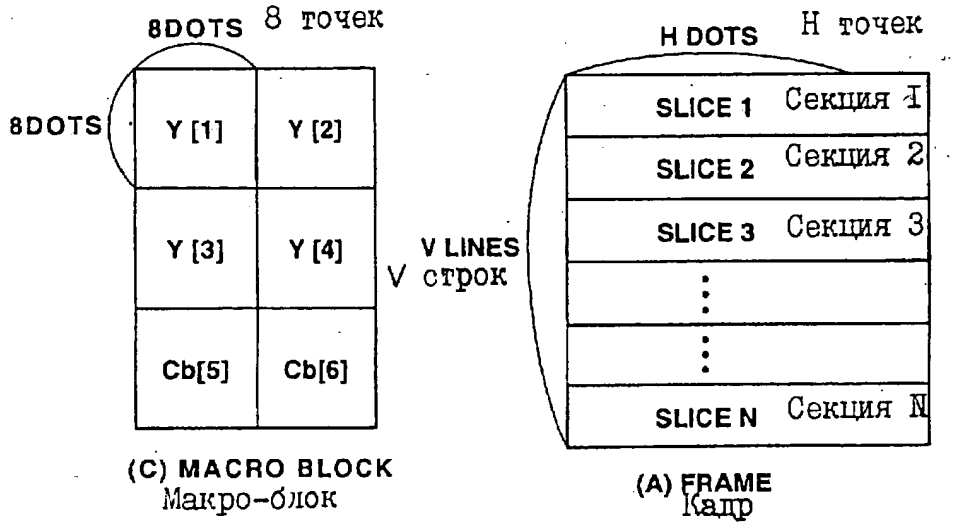
Фиг.8

RU 2123769 C1

RU 2123769 C1



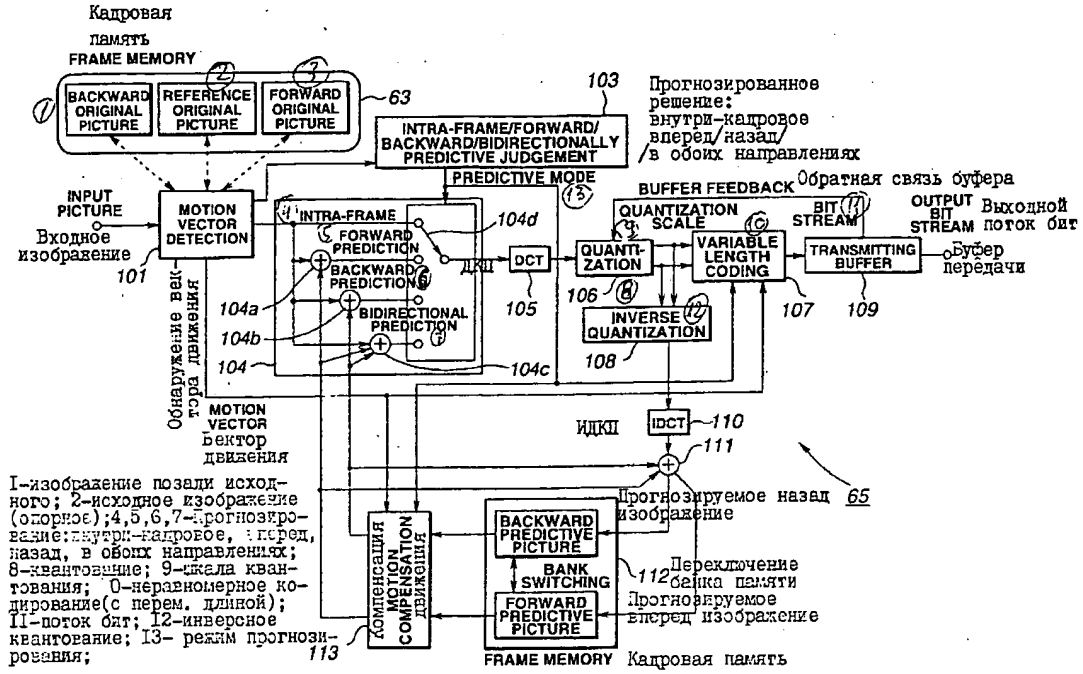
Фиг.9



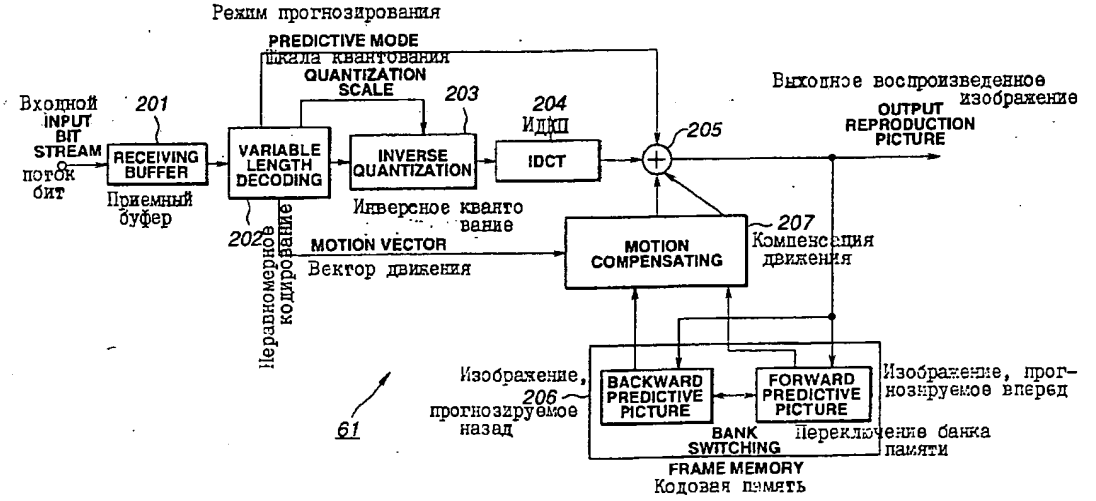
Фиг.10

RU 2123769 C1

RU 2123769 C1



Фиг.11



Фиг.12

RU 2123769 C1

RU 2123769 C1

OIPE IAP55
OCT 22 2007
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,996	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on October 17, 2007.

By: Rennae Johnson

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

These items were also first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached document do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

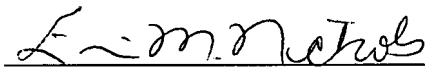
No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: October 17, 2007

By: 
Erin M. Nichols
Reg. No. 57,125

OIPE
OCT 22 2007
PATENT & TRADEMARK OFFICE

FORM 1449* (IAP55) INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5,144,426	09.01.1992	Tanaka et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	01/84850	8 Nov. 2001	WO			NA	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 November 2001 (08.11.2001)

PCT

(10) International Publication Number
WO 01/84850 A1

(51) International Patent Classification[7]: H04N 7/50

(21) International Application Number: PCT/EP01/04319

(22) International Filing Date: 13 April 2001 (13.04.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
00201505.5 27 April 2000 (27.04.2000) EP
00203828.9 2 November 2000 (02.11.2000) EP

(71) Applicant: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(72) Inventors: BRULS, Wilhelmus, H., A.; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). KLEIN GUNNEWIEK, Reinier, B., M.; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(74) Agent: STEENBEEK, Leonardus, J.; Internationaal Octrooibureau B.V., Prof Holstlaan 6, NL-5656 AA Eindhoven (NL).

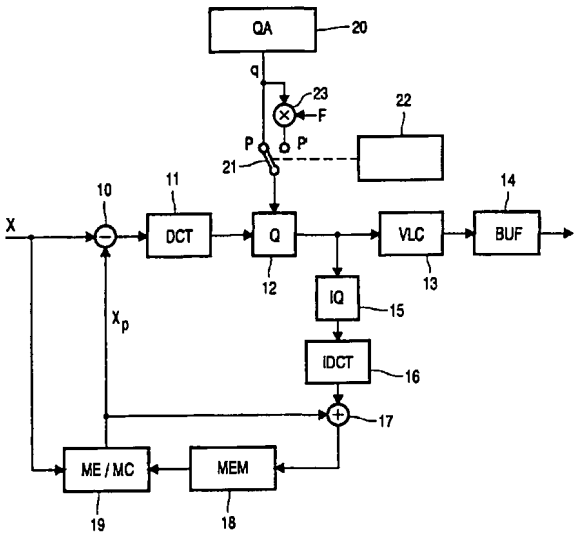
(81) Designated States *(national)*: CN, JP, KR.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: VIDEO COMPRESSION



(57) Abstract: The concept of B-frames gives the MPEG video compression standard its high encoding efficiency. However, B-frame encoding roughly doubles the complexity of an MPEG encoder. In view thereof, MPEG encoders have been developed which produce I-frames and P-frames only. They are less complex but also less efficient. To improve the efficiency of such "IPP encoders", selected P-frames are quantized more coarsely than other P-frames, for example, by multiplying the conventional quantization step size by 1.4. Although this results in isolated frames ("virtual B-frames") being encoded with a lower quality, the overall perceptual quality is not affected. It has been found that the gain in bit rate obtained by the coarser quantization is not lost in subsequent P-frames, even though the subsequent frames are encoded with reference to the lower quality frames.

WO 01/84850 A1

Video compression

FIELD OF THE INVENTION

The invention relates to a method of compressing a video signal, the method comprising predictively encoding frames of said video signal with reference to a prediction frame, calculating a quantization parameter for each encoded frame, and quantizing the encoded frames in accordance with said quantization parameter. The invention also relates to a compression arrangement, to a transmission or recording method and arrangement, to the compressed video signal and to a storage medium comprising that signal.

BACKGROUND OF THE INVENTION

A video compression method as defined in the opening paragraph has been standardized by the Motion Frames Expert Group and is well-known as MPEG1 or MPEG2. The known method includes transformation of video pixels into frequency coefficients, quantization of said coefficients, and variable-length coding of the quantized coefficients. The quantization is controlled so as to achieve a desired quality or bit rate of the compressed signal.

The MPEG compression method produces I, P and B-frames. I-frames are encoded autonomously, i.e. without reference to another frame. P-frames are predictively encoded with reference to a previous (possibly motion-compensated) I or P-frame. B-frames are bidirectionally predictively encoded with reference to a previous and a subsequent I or P frame. B-frames are not themselves used as reference for encoding other frames.

The concept of B-frames in MPEG provides maximum encoding efficiency. However, the use of B-frames roughly doubles the complexity, memory capacity and memory bandwidth. In view thereof, MPEG encoders have been developed which produce I and P-frames only ("IP encoders"). A disadvantage of IP encoders is their efficiency. They need approximately 10-20% more bit rate than IPB encoders.

OBJECT AND SUMMARY OF THE INVENTION

It is an object of the invention to provide an arrangement and method which overcomes the above-mentioned disadvantage of prior-art IP encoders.

To this end, the invention provides a video compression arrangement and method, a compressed signal, a storage medium, and a transmission or recording method and arrangement as defined in the independent claims. Advantageous embodiments are defined in the dependent claims.

The method in accordance with the invention quantizes selected P-frames more coarsely than other P-frames. This reduces the bit cost but degrades the image quality of said frames. The invention has a surprising effect. It was expected that the corresponding gain in bit cost would be lost in subsequent P-frames because the lower quality frames are used as prediction for subsequent P-frames. However, experiments have shown that this is not the case. It has been found that an IPPPP.. sequence of frames, in which the quantization parameter of every other P-frame is multiplied by a factor of 1.4, has substantially the same bit rate as a conventional IBPBP.. sequence having the same perceptual visual quality. In view thereof, the lower quality P-frames are also referred to as "virtual B-frames".

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a schematic diagram of a preferred embodiment of an arrangement for compressing a video signal encoder in accordance with the invention;

Figs. 2A and 2B show diagrams illustrating the performance of the arrangement in accordance with the invention compared with the performance of a prior-art arrangement;

Fig. 3 shows a block diagram of embodiments of arrangements for transmitting and receiving a video signal; and

Fig. 4 shows a block diagram of embodiments of arrangements for recording a video signal on a storage medium and for playing back from the storage medium.

DESCRIPTION OF EMBODIMENTS

Fig. 1 shows a schematic diagram of an MPEG encoder in accordance with the invention. The Figure shows the encoder in the state in which P-frames are encoded. The encoder is a conventional MPEG encoder in the sense that it comprises a subtraction circuit **10**, a discrete cosine transformer (DCT) **11**, a quantizer (Q) **12**, a variable-length coder (VLC) **13**, a buffer (BUF) **14**, an inverse quantizer (iQ) **15**, an inverse discrete cosine transformer (iDCT) **16**, an adder **17**, a frame memory (MEM) **18**, a motion estimation and compensation circuit (ME/MC) **19**, and a quantization adapter (QA) **20**.

Briefly summarized, the known encoder operates as follows. The input video frame X is divided into blocks of 8×8 pixels. The difference between each pixel block of input frame X and the corresponding block of a prediction frame X_p is discrete cosine transformed into a block of 8×8 coefficients. The coefficients are subsequently quantized, by which perceptually irrelevant picture details are irreversibly removed (lossy compression). The quantized coefficients are variable-length encoded and stored in a buffer from which the signal is applied to a transmission channel or record carrier. The encoded frame is locally decoded by inverse quantization, inverse discrete cosine transformation, and addition to the prediction frame X_p. The reconstructed frame is stored in the frame memory and subjected to motion estimation and compensation so as to constitute the prediction frame for the next input frame.

The encoder includes a quantization adapter **20** for calculating the quantization steps with which the DCT-coefficients are quantized. In this embodiment, the MPEG2 quantization mechanism is used in which a predetermined quantization matrix, which defines the step sizes to be applied to the respective coefficients of an 8×8 coefficient block, is multiplied by a quantization scale factor q (herein further referred to as quantization parameter). The quantization parameter is adapted from frame to frame, but may be 'modulated' within a frame as a function of local image details. The quantization parameter may be controlled to represent a given image quality (resulting in a variable bit rate) or a given bit rate (resulting in a variable quality). Various embodiments of quantization adapters (also referred to as bit rate controllers) are known in the art and may be employed in the encoder according to the invention.

The arrangement in accordance with the invention increases the quantization parameter q for selected frames, thereby degrading the image quality of said frames but reducing their bit costs. In this embodiment, the arrangement includes a multiplier **23** which multiplies the quantization parameter q calculated by the quantization adapter **20** by a predetermined factor F (e.g. F=1.4). A switch **22** has a position P in which the conventional quantization parameter q is applied to the quantizer **12** and a position P' in which the coarser quantization parameter F.q is applied to the quantizer. The switch is controlled by a control circuit **22** in a predetermined manner. For example, the control circuit selects every other P-frame to be more coarsely quantized.

Fig. 2A shows a diagram illustrating the performance of a conventional MPEG2 encoder which produces a stream of IPPP.. frames (no B-frames). Each frame is quantized in accordance with the quantization parameter q as calculated by the quantization

adapter **20**. The lower row of figures denotes the bit cost of the respective frame, expressed as a percentage of the bit cost of the respective I-frame. The bit cost of P-frames appears to be 38% in this example.

Fig. 2B shows a similar diagram for an encoder in accordance with the invention. The quantization adapter **20** has been set to produce the same image quality as in Fig. 2A. In accordance therewith, the bit cost for I-frames is the same as in Fig. 2A. Every other P-frame (denoted P' in the Figure) is now quantized with the quantization parameter 1.4q. The bit cost of the P'-frames is thereby reduced from 38% to 26%. The image quality of said frames is reduced in proportion therewith. The surprising effect of the invention is that the gain in bit cost is not lost in the subsequent 'conventional' P-frames. As shown in Fig. 2B, the bit cost of 'conventional' P-frames increases only from 38% to 42%. The net result is a considerable reduction of the bit rate at the same perceptual image quality (or a higher perceptual quality at the same bit rate) of the encoded video stream. In a practical experiment, the bit rate of a typical video signal was reduced from 15.2 Mbit/sec to 12.9 Mbit/sec at the same perceptual quality.

It is to be noted that the bit stream produced by an MPEG encoder in accordance with the invention fully complies with the MPEG standard. It should also be noted that although the invention has been described with reference to an IPP.. encoder (no B-frames), the invention does not exclude B-frame encoding. For example, an encoder may produce an IBPBP.. sequence in which selected P-frames have been quantized with the coarser quantization parameter. The coarser quantization parameter may even be applied to I-frames to the extent that such I-frames are used as prediction frames for subsequent frames.

Fig. 3 shows embodiments of arrangements for transmitting and receiving a video signal. An encoder 100 receives an image signal I on input 102. The encoder 100 is preferably constructed in accordance with the embodiment of Fig. 1, but may be any kind of compressed video signal encoder that supplies compressed video signals including first quantization parameters (q) representing a first quality or bit rate for quantizing selected first frames (P) of predictively encoded frames, and second quantization parameters (F.q) representing a second quality or bit rate that is lower than the first quality or bit rate for quantizing selected second frames (P') of the video signal. The compressed video signal is supplied at output 106 to a transmitter 108. The transmitter 108 then converts the compressed video signal into a transmission signal and applies the transmission signal to a transmission medium 110. The transmitter 108 and the transmission medium 110 may take any known form, for example, in the case of a broadcast, the transmitter 108 modulates the compressed

video signal onto a radio frequency (RF) carrier wave, and the transmission medium 110 may comprise airwaves containing the radio waves, or a cable carrying the radio waves. Alternatively, it is known to supply digital signals via the Internet. As such, the transmitter 108 may comprises means for transmitting the compressed video signal over the transmission medium 110 comprising the Internet.

A receiver 112 coupled to the transmission medium 110 then receives the transmission signal and applies the image signal to a display device 118.

Fig. 4 shows a block diagram of embodiments of arrangements for recording and playing back a compressed video signal onto and from a record carrier. This arrangement is substantially similar to that shown in Fig. 3, except that the compressed video signal at the output 106 of the encoder 100 is applied to a recording device 120. The recording device 120 then converts the compressed video signal into a recording signal and records the recording signal onto the record carrier 122. The recording device 120 and the record carrier 122 may take any known form. For example, in the case of the record carrier 122 being a magnetic video tape, the recording device 120 may take the form of a helical scan video tape recorder. Alternatively, the record carrier 122 may be an optical disk, for example, CD-ROM, CD-R, DVD, DVD-ROM, DVD-R/W, etc. In that event, the recording device 120 would take the form of an optical disk recorder.

In order to play back the recording signal RS, the record carrier 122 is inserted into a playback device 124 which processes the recording signal RS and generates an image signal. As with the embodiment of Fig. 3, the playback device 124 applies the image signal to the display device 118.

The invention can be summarized as follows. The concept of B-frames gives the MPEG video compression standard its high encoding efficiency. However, B-frame encoding roughly doubles the complexity of an MPEG encoder. In view thereof, MPEG encoders have been developed which produce I-frames and P-frames only. They are less complex but also less efficient. To improve the efficiency of such "IPP encoders", selected P-frames are quantized more coarsely than other P-frames, for example, by multiplying the conventional quantization step size by 1.4. Although this results in isolated frames ("virtual B-frames") being encoded with a lower quality, the overall perceptual quality is not affected. It has been found that the gain in bit rate obtained by the coarser quantization is not lost in subsequent P-frames, even though the subsequent frames are encoded with reference to the lower quality frames.

It should be noted that the above-mentioned embodiments illustrate rather than limit the invention, and that those skilled in the art will be able to design many alternative embodiments without departing from the scope of the appended claims. In the claims, any reference signs placed between parentheses shall not be construed as limiting the claim. The word "comprising" does not exclude the presence of elements or steps other than those listed in a claim. The word "a" or "an" preceding an element does not exclude the presence of a plurality of such elements. The invention can be implemented by means of hardware comprising several distinct elements, and by means of a suitably programmed computer. In the device claim enumerating several means, several of these means can be embodied by one and the same item of hardware. The mere fact that certain measures are recited in mutually different dependent claims does not indicate that a combination of these measures cannot be used to advantage.

CLAIMS:

1. A method of compressing a video signal, the method comprising:

predictively encoding (10,11) frames (X) of said video signal with reference to a prediction frame (X_p),

calculating (20) a quantization parameter (q) for each encoded frame,

quantizing (12) the encoded frames in accordance with said quantization parameter,

characterized in that said step of calculating the quantization parameter includes calculating a first quantization parameter (q) representing a first quality or bit rate for quantizing selected first frames (P) of said predictively encoded frames, and a second quantization parameter (F.q) representing a second quality or bit rate that is lower than said first quality or bit rate for quantizing selected second frames (P') of the video signal, the method further including:

decompressing (15-18) the compressed second frames to constitute the prediction frame (X_p) for predictively encoding the first frames.

2. A method as claimed in claim 1, wherein the step of calculating the second quantization parameter includes calculating said first quantization parameter (q) and multiplying (23) said first quantization parameter by a given factor (F).

3. A method as claimed in claim 1, wherein said predictively encoded frames constitute a series of successive frames, the second selected frames being every other frame of said series.

4. An arrangement for compressing a video signal, the arrangement comprising:

encoding means (10,11) for predictively encoding frames (X) of said video signal with reference to a prediction frame (X_p),

calculation means (20) for calculating a quantization parameter (q) for each encoded frame,

a quantizer (12) for quantizing the encoded frames in accordance with said quantization parameter,

characterized in that said calculation means (20) are arranged to calculate a first quantization parameter representing a first quality or bit rate for quantizing selected first frames (P) of said predictively encoded frames, and a second quantization parameter (F.q) representing a second quality or bit rate that is lower than said first quality or bit rate for quantizing selected second frames (P') of the video signal, the arrangement further including:

means (15-18) for decompressing the compressed second frames to constitute said prediction frame (X_p) for predictively encoding first selected frames.

5. An arrangement as claimed in claim 4, wherein said calculation means (20) comprise a multiplier (23) for multiplying the first quantization parameter (q) by a given factor (F).

6. An arrangement as claimed in claim 4, wherein said predictively encoded frames constitute a series of successive frames, the second selected frames being every other frame of said series.

7. A compressed video signal, comprising:

a prediction frame (X_p),

predictively encoded (10,11) frames (X) that have been predictively encoded with reference to the prediction frame (X_p),

respective quantization parameters (q) for respective encoded frames, the encoded frames having been quantized (12) in accordance with said respective quantization parameters, the quantization parameters including first quantization parameters (q) representing a first quality or bit rate for quantizing selected first frames (P) of said predictively encoded frames, and second quantization parameters (F.q) representing a second quality or bit rate that is lower than said first quality or bit rate for quantizing selected second frames (P') of the video signal.

8. A storage medium (122) on which the compressed video signal of claim 7 has been stored.

9. A method of transmitting or recording a video signal, the method comprising:

generating the compressed video signal of claim 7; and

transmitting or storing the compressed video signal.

10. An arrangement for transmitting or recording a video signal, the arrangement comprising:

means (100) for generating the compressed video signal of claim 7; and

means (108, 120) for transmitting or recording the compressed video signal.

1/2

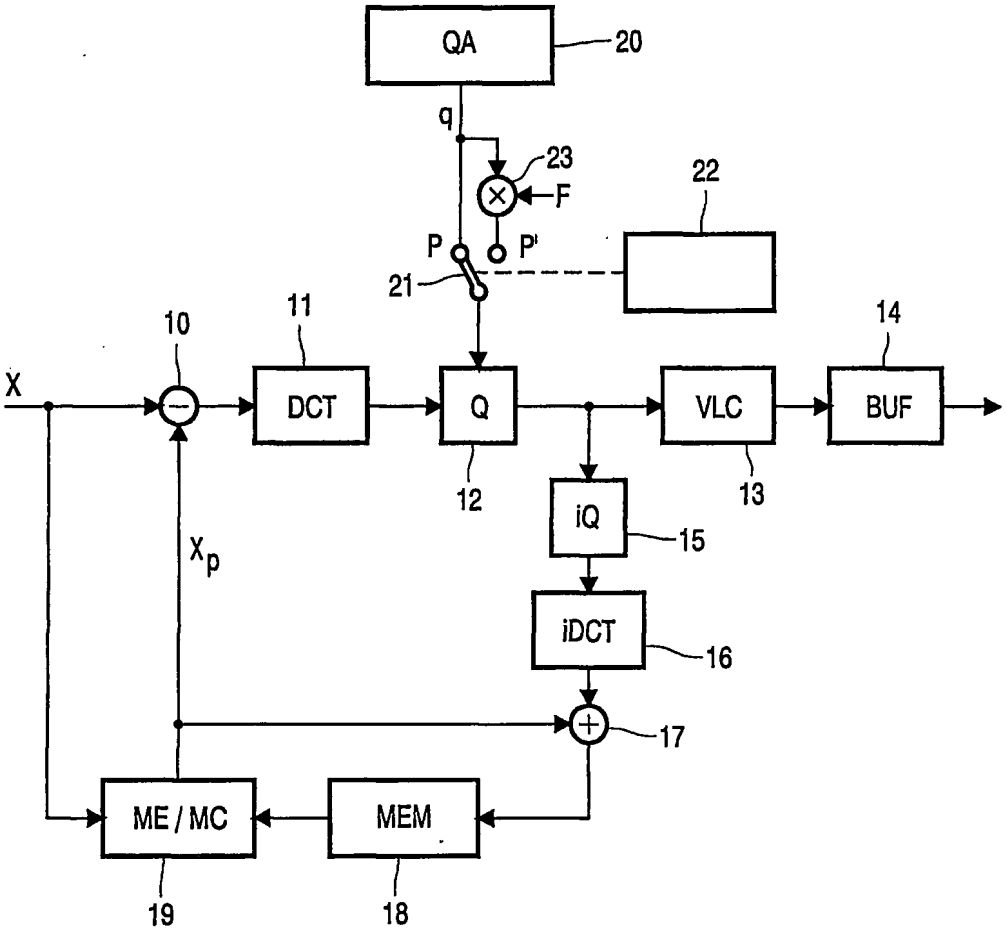


FIG. 1

I	P	P	P	P	P	I	P	P	P	P	P
100%	38%	38%	38%	38%	38%	100%	38%	38%	38%	38%	38%

FIG. 2A

I	P'	P	P'	P	P'	I	P'	P	P'	P	P'
100%	26%	42%	26%	42%	26%	100%	26%	42%	26%	42%	26%

FIG. 2B

2/2

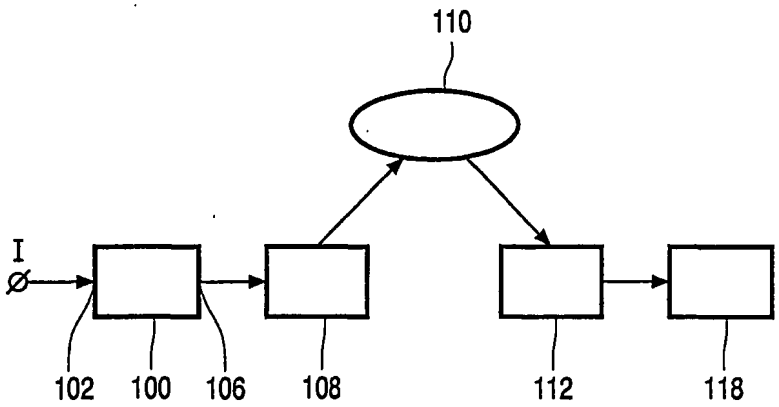


FIG. 3

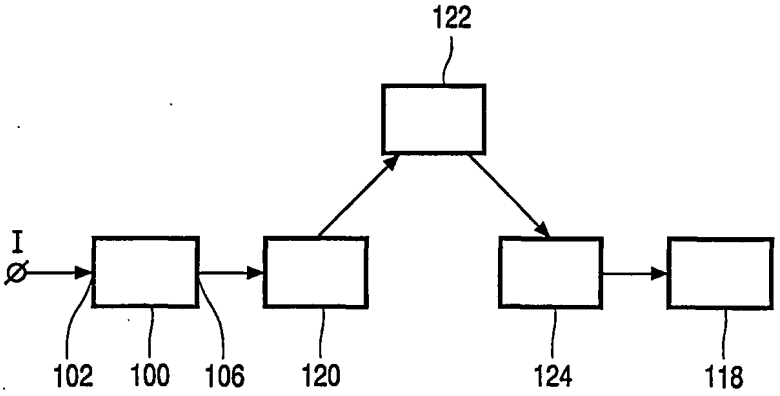


FIG. 4

INTERNATIONAL SEARCH REPORT

International Application No
I/EP 01/04319

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N7/50

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
INSPEC, EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WENGER S: "Temporal scalability using P-pictures for low-latency applications" 1998 IEEE SECOND WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING (CAT.NO.98EX175), 1998 IEEE SECOND WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING, 7 - 9 December 1998, pages 559-564, XP002177102 Redondo Beach, CA, USA the whole document	1-10
A	US 5 832 121 A (ANDO YUJI) 3 November 1998 (1998-11-03) column 11, line 64 -column 14, line 11	1,4,7,9, 10

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 September 2001	24/09/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Marie-Julie, J-M

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

National Application No
T/EP 01/04319

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5832121 A	03-11-1998	JP 8098185 A	12-04-1996
		EP 1098530 A	09-05-2001
		EP 0705041 A	03-04-1996
		US 5757968 A	26-05-1998

Form PCT/ISA/210 (patent family annex) (July 1992)

OIPE MAP76
JUN 16 2008
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,996	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on June 10, 2008.

By: Rennae Johnson

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the item of information listed on the enclosed Form 1449 is brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the item listed on the enclosed Form 1449.

This item was also first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.

Applicant has not provided a copy of the U.S. publication/patent listed on the enclosed Form 1449 in accordance with 37 C.F.R. § 1.98(a)(2).

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to

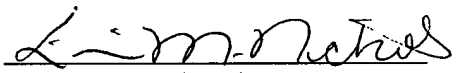
establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: June 10, 2008

By: 
Erin M. Nichols
Reg. No. 57,125

Date Mailed: June 10, 2008

O I P E IAP76 JUN 16 2008 PATENT & TRADEMARK OFFICE

FORM 1449*	INFORMATION DISCLOSURE STATEMENT	Docket Number: TPO.004.C1	Application Number: 11/338,996
	IN AN APPLICATION	Applicant: HANNUKSELA	
	(Use several sheets if necessary)	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	20010024472	09/27/2001	Sporer et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
11/338,996		2621	

OC000000032242968

Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 76385 on 09/23/2008

- **Correspondence Address**
- **Power of Attorney Address**

The address of record for Customer Number 76385 is:

76385
Hollingsworth & Funk, LLC
8009 34th Avenue South
Suite 125
Minneapolis, MN 54425

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,996	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being transmitted to the U.S. Patent and Trademark Office via the Office electronic filing system in accordance with 37 CFR 1.6(a)(4) on March 11, 2009.

By: /Tracey M. Dotter/
Tracey M. Dotter

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

These items were also first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), copies of the foreign references listed on the enclosed Form 1449 are provided. As indicated in the communication from the Chinese patent office (a copy with an English translation is attached) the English-language EP publication corresponds to the cited Chinese reference.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: March 11, 2009

By: /Erin M. Nichols/
Erin M. Nichols
Reg. No. 57,125

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	EP 0798932	10/1/1997	EP			NA	
	CN 1173783A	2/18/1998	China				X

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

中华人民共和国国家知识产权局

100022 北京市朝阳区东三环中路39号建外SOHO A座31层 北京市金杜律师事务所 王茂华,赵林琳	发文日
申请号:200610093533	
申请人:诺基亚有限公司	
发明名称:视频编码中的图像帧编组	

专利局 2008.02.13 收文

第 2 次审查意见通知书

1. ☑审查员已收到申请人于2008年10月13日提交的意见陈述书,在此基础上审查员对上述专利申请继续进行实质审查。

☐根据国家知识产权局专利复审委员会于 年 月 日作出的复审决定,审查员对上述专利申请继续实质审查。

☐_____

2. ☐申请人于 年 月 日提交的修改文件,不符合专利法实施细则第51条第3款的规定。

3. 继续审查是针对下述申请文件进行的:

☐上述意见陈述书中所附的经修改的申请文件。

☑前次审查意见通知书所针对的申请文件以及上述意见陈述书中所附的经修改的申请文件替换页。

☐前次审查意见通知书所针对的申请文件。

☐上述复审决定所确定的申请文件。

☐_____

4. ☐本通知书未引用新的对比文件。

☑本通知书引用下述对比文件(其编号续前,并在今后的审查过程中继续沿用):

编号	文件号或名称	公开日期(或抵触申请的申请日)
1	CN1173783A	1998.2.18

5. 审查的结论性意见:

☐关于说明书:

☐申请的内容属于专利法第5条规定的不授予专利权的范围。

☐说明书不符合专利法第26条第3款的规定。

☐说明书的修改不符合专利法第33条的规定。

☐说明书的撰写不符合专利法实施细则第18条的规定。

☐_____

☑关于权利要求书:

☑权利要求 1,10 不具备专利法第22条第2款规定的新颖性。

☐权利要求_____不具备专利法第22条第3款规定的创造性。

☐权利要求_____不具备专利法第22条第4款规定的实用性。

☐权利要求_____属于专利法第25条规定的不授予专利权的范围。

☐权利要求_____不符合专利法第26条第4款的规定。

☐权利要求_____不符合专利法第31条第1款的规定。

☐权利要求_____的修改不符合专利法第33条的规定。

☐权利要求_____不符合专利法实施细则第2条第1款的规定。

☐权利要求_____不符合专利法实施细则第13条第1款的规定。

中华人民共和国国家知识产权局 专利审查业务章 104

☑权利要求 1, 9-10 不符合专利法实施细则第 20 条的规定。
☐权利要求_____不符合专利法实施细则第 21 条的规定。
☐权利要求_____不符合专利法实施细则第 22 条的规定。
☑权利要求4-6, 8-9, 13-15, 17不符合专利法实施细则第 23 条的规定。
☐_____
☐分案的申请不符合专利法实施细则第 43 条第 1 款的规定。
上述结论性意见的具体分析见本通知书的正文部分。

6. 基于上述结论性意见，审查员认为：
☐申请人应按照通知书正文部分提出的要求，对申请文件进行修改。
☑申请人应在意见陈述书中论述其专利申请可以被授予专利权的理由，并对通知书正文部分中指出的不符合规定之处进行修改，否则将不能授予专利权。
☐专利申请中没有可以被授予专利权的实质性内容，如果申请人没有陈述理由或者陈述理由不充分，其申请将被驳回。
☐_____

7. 申请人应注意下述事项：
(1) 根据专利法第 37 条的规定，申请人应在收到本通知书之日起的贰个月内陈述意见，如果申请人无正当理由逾期不答复，其申请将被视为撤回。
(2) 申请人对其申请的修改应符合专利法第 33 条和实施细则第 51 条的规定，修改文本应一式两份，其格式应符合审查指南的有关规定。
(3) 申请人的意见陈述书和/或修改文本应邮寄或递交国家知识产权局专利局受理处，凡未邮寄或递交给受理处的文件不具备法律效力。
(4) 未经预约，申请人和/或代理人不得前来国家知识产权局专利局与审查员举行会晤。

8. 本通知书正文部分共有 2 页，并附有下述附件：
☑引用的对比文件的复印件共 1 份 37 页。
☐_____

审查员：周学斌(976H)
2009 年 1 月 21 日

审查部门 审查协作中心

第二次审查意见通知书正文

申请号：2006100093533

申请人于2008年10月13日提交了意见陈述书和经过修改的申请文件，审查员在阅读了上述文件后，对本申请继续进行审查，再次提出如下审查意见。

1.权利要求1请求保护一种用于解码压缩视频序列的方法，对比文件1（CN1173783A）公开了一种解码器，同时也公开了该解码器具体的解码过程，具体披露了以下技术特征（参见对比文件1说明书第6页第1行-第12行）：接收传输数据流，接收其中的一个不连续指示位（相当于该权利要求的不连续性的指示），将传输包的有效内容（相当于该权利要求的不连续性相应的多个图像帧）存入存储器（相当于该权利要求的缓冲存储器），通过解码器对存储器中的数据进行处理。可见，该权利要求所要求保护的技术方案与该对比文件所公开的内容相比，所不同的仅仅是文字表达方式上略有差别，其技术方案实质上是相同的，且两者属于相同的技术领域，并且采用相同的技术方案解决了同样的技术问题，并能产生相同的技术效果，因此该权利要求所要求保护的技术方案不具备专利法第二十二条第二款规定的新颖性。

2.权利要求10请求保护一种用于解码压缩视频序列的视频解码器，对比文件1（CN1173783A）公开了一种解码器，，具体披露了以下技术特征（参见对比文件1说明书第6页第1行-第12行）：接收传输数据流，接收其中的一个不连续指示位（相当于该权利要求的不连续性的指示），将传输包的有效内容（相当于该权利要求的不连续性相应的多个图像帧）存入存储器（相当于该权利要求的缓冲存储器），解码器与存储器相连，通过解码器对存储器中的数据进行处理。可见，该权利要求所要求保护的技术方案与该对比文件所公开的内容相比，所不同的仅仅是文字表达方式上略有差别，其技术方案实质上是相同的，且两者属于相同的技术领域，并且采用相同的技术方案解决了同样的技术问题，并能产生相同的技术效果，因此该权利要求所要求保护的技术方案不具备专利法第二十二条第二款规定的新颖性。

3.权利要求1和10中的出现了特征“所述图像帧”（参见上述权利要求的最后一行），在该特征之前多次出现过特征“图像帧”，并且在不同的步骤中，“图像帧”包含的内容也是不同的，因此，对该特征的引用是不清楚的，不能明确是引用那个部分的“图像帧”，因此导致了该权利要求的保护范围不清楚，不符合专利法实施细则第二十条第一款的规定。

4.权利要求9中的特征“参考”含义不清楚，不能确定参考数据的类型，因此导致该权利要求的保护范围不清楚，不符合专利法实施细则第二十条第一款的规定。

5.权利要求4-6，8-9，13-15，17本身是多项从属权利要求，其引用的权利要求中还包括多项从属权利要求，因此不符合专利法实施细则第二十三条第二款的规定。申请人应当对该权利要求的引用关系进行修改。

6.权利要求10，12，14-15，17-18，20中，在"还被安排来"之后没有任何标点，申请人应该对其进行修改、

基于上述理由，本发明专利申请按照目前的文本不能被授权。申请人应在本通知指定的答复期限内提交新修改的权利要求书（请全文替换并附上带有修改标记的修改草稿）。请在意见陈述书中详细论述已克服上述缺陷的理由。对申请文件的修改应当符合专利法第三十三条的规定，不得超出原说明书和权利要求书记载的范围。对申请文件进行的任何改动，务请详细说明修改没有超范围的理由。

审查员：周学斌

代码：976H

THE PATENT OFFICE OF THE PEOPLE'S REPUBLIC OF CHINA

Address: 6 Xi Tu Cheng Lu, Haidian, Beijing 100088

Applicant:	**NOKIA CORPORATION**	
Attorney:	**Maohua WANG**	Date of Notification:
Application No.:	**200610009353.3**	Date13Month02Year2009
Title of the Invention:	**GROUPING OF IMAGE FRAMES IN VIDEO CODING**	

Notification of 2nd Office Action

1. ☒ The examiner received the response submitted by the applicant on_____ to the1stOffice Action and further examination as to substance has been carried out on the above-identified patent application for invention on this new basis.
   ☐ According to the Reexamination Decision made by the Patent Reexamination Board of the Patent Office on _____, examination as to substance on the above-identified application has been resumed.

2. Further examination as to substance has been carried out based on the documents as specified below:
   ☐ The amended application documents attached to the response to the previous Office Action.
   ☒ The application documents based on which the previous examination was carried out and the substitution pages attached to the response to the previous Office Action.
   ☐ The application documents based on which previous examination was carried out.
   ☐ The application documents confirmed by the Reexamination Decision.

3. ☐ No further reference documents are cited in this Office Action.
   ☒ Below is/are the reference document(s) cited in this Notification:

No.	Number(s) or Title(s) of Reference(s)	Date of Publication(or the filing date of conflicting application)
1	CN1173783A	Date:18Month:2Year: 1998
		Date: Month: Year:
		Date: Month: Year:
		Date: Month: Year:
		Date: Month: Year:

4. Conclusions of the Action:
   ☐ On the Specification:
   ☐ The amendments to the description do not comply with Article 33 of the Patent Law.
   ☐ The subject matter contained in the application is not patentable under Article 5 of the Patent Law.
   ☐ The description does not comply with Article 26 paragraph 3 of the Patent Law.
   ☐ The draft of the description does not comply with Rule 18 of the Implementing Regulations.

   ☒ On the Claims:

☐ The amendments to claims _____do not comply with Article 33 of the Patent Law.
☐ Claim(s)_____is/are not patentable under Article 25 of the Patent Law.
☐ Claim(s)_____does/do not comply with the definition of inventions prescribed by Rule 2 paragraph 1 of the Implementing Regulations.
☒ Claim(s) 1, 10 does/do not possess the novelty as required by Article 22 paragraph 2 of the Patent Law.
☐ Claim(s)_____does/do not possess the inventiveness as required by Article 22 paragraph 3 of the Patent Law.
☐ Claim(s)_____does/do not possess the practical applicability as required by Article 22 paragraph 4 of the Patent Law.
☐ Claim(s)_____does/do not comply with Article 26 paragraph 4 of the Patent Law.
☐ Claim(s)_____does/do not comply with Article 31 paragraph 1 of the Patent Law.
☒ Claim(s) 1, 9-10 does/do not comply with the provisions of Rules 20 of the Implementing Regulations.
☒ Claim(s) 4-6, 8-9, 13-15, 17 does/ do not comply with the provisions of Rules 23 of the Implementing Regulations..
☐ Claim(s) _____does/do not comply with the provisions of Rule 12 paragraph 1 of the Implementing Regulations.

The detailed explanation of the above conclusions is set forth in the text portion of the Notification.

5. In view of the conclusions set forth above, the Examiner is of the opinion that:
☐ The applicant should make amendments to the application documents as directed in the text portion of the Notification.
☒ The applicant should expound in the response reasons why the application is patentable and make amendments to the application where there are deficiencies as pointed out in the text portion of the Notification, otherwise, the application will be rejected.
☐ The application contains no allowable invention, and therefore, if the applicant fails to submit sufficient reasons to prove that the application does have merits, it will be rejected.
6. The followings should be taken into consideration by the applicant in making the response:
(1) Under Article 37 of the Patent Law, the applicant should respond to the office action within 2 months counting from the date of receipt of the Notification. If, without any justified reason, the time limit is not met, the application shall be deemed to have been withdrawn.
(2) Any amendment to the application should be in conformity with the provisions of Article 33 of the Patent Law and Rule 51 of the Implementing Regulations. Substitution pages should be in duplicate and the format of the substitution should be in conformity with the relevant provision contained in "The Examination Guidelines".
(3) The response to the Notification and/or revision of the application should be mailed to or handed over to the "Reception Division" of the Patent Office, and documents not mailed or handed over to the Reception Divisions have no legal effect.
(4) Without an appointment, the applicant and/or his agent shall not interview with the Examiner in the Patent Office.
7. This Notification contains a text portion of 2 pages and the following attachments:
☒ 1cited reference(s), totaling 37pages.

Examination Dept. 9 Examiner: 976H Seal of the Examination Department

Text of the Second Office Action

The applicant submitted the observations and amended application documents on October 13, 2008. After reading the above documents, the examiner continues examination on the present application and provides the following opinions:

1. Claim 1 is directed to a method for decoding a compressed video sequence. 1 (CN1173783A, corresponding to EP0798932A2 of the same patent family) discloses a decoder, and meanwhile discloses concrete decoding process of the decoder, and specifically discloses the following technical features (see page 5, lines 20-30 of the description, EP0798932A2): receiving transport data stream, receiving a discontinuity indicator (equivalent to discontinuity indication of claim 1) thereof, storing the transport packet payloads (equivalent to the number of image frames corresponding to a discontinuity of claim 1) in memory (equivalent to the buffer memory of claim 1), and processing by the decoder the data in the memory. It is seen that the technical solution for which protection is sought in the claim differs from the contents disclosed by D1 only slightly in difference, and their technical solutions are substantively identical. Further, claim 1 and D1 fall within the same technical field, solve the same technical problem by adopting the same technical solution and can achieve the same technical effect. Thus, the technical solution for which protection is sought in claim 1 lacks novelty under Article 22.2 of the Chinese Patent Law.

2. Claim 10 is directed to a video decoder for decoding a compressed video sequence. 1 (CN1173783A, corresponding to EP0798932A2 of the same patent family) discloses a decoder, and meanwhile discloses concrete decoding process of the decoder, and specifically discloses the following

technical features (see page 5, lines 20-30 of the description, EP0798932A2): receiving transport data stream, receiving a discontinuity indicator (equivalent to discontinuity indication of claim 10) thereof, storing the transport packet payloads (equivalent to the number of image frames corresponding to a discontinuity of claim 10) in memory (equivalent to the buffer memory of claim 10) connected to the decoder, and processing by the decoder the data in the memory. It is seen that the technical solution for which protection is sought in the claim differs from the contents disclosed by D1 only slightly in difference, and their technical solutions are substantively identical. Further, claim 10 and D1 fall within the same technical field, solve the same technical problem by adopting the same technical solution and can achieve the same technical effect. Thus, the technical solution for which protection is sought in claim 10 lacks novelty under Article 22.2 of the Chinese Patent Law.

3. In claims 1 and 10, there recites the wording "the image frames" (see the last line of the above claims). Since the feature "image frames" have been previously mentioned for several times, and in different steps, the contents contained in the "image frames" are different, citation of the feature is unclear and it is unclear which "image features" are cited. Thus the respective protection scope of the claims is unclear in the sense of Rule 20.1 of the Implementing Regulations of the Chinese Patent Law.

4. In claim 9, the wording "a reference" is unclear in meaning and it is unable to determine the type of reference data, which renders the protection scope of claim 9 unclear in the sense of Rule 20.1 of the Implementing Regulations of the Chinese Patent Law.

5. Claims 4-6, 8-9, 13-15 and 17 per se are multiple dependent claims, while

the claims to which they refer also contain multiple dependent claims, which therefore do not comply with Rule 23.2 of the Implementing Regulations of the Chinese Patent Law. Please be advised to amend the dependencies of the claims.

6. (A formality issue is omitted here, which will be attended to by this Attorney without your further input.)

To sum up, the present application can not be granted a patent right under the current application documents. The applicant should submit newly amended claims (please replace the whole text attached with the amended draft with marks) within the time limit for response as prescribed in the Office Action. Please state in detail the reasons why the above defects have been overcome in the observations. It should note that subject to Article 33 of the Chinese Patent Law, any amendments to the application documents should not go beyond the scope of the disclosure contained in the original description and claims. For any change to the application documents, please state reasons in detail why such amendments do not go beyond the original scope.

Examiner: Zhou Xuebin
Code:976H

Electronic Acknowledgement Receipt

EFS ID:	4945493
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	11-MAR-2009
Filing Date:	25-JAN-2006
Time Stamp:	18:05:22
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Foreign Reference	EP0798932.pdf	1101668 1b1772a84e65bbd73e5edcc39386750838 60bb8c	no	22

Warnings:	
Information:	

2	Foreign Reference	CN1173783A.pdf	1780503 2bbcb9681463a00d81a20d40bac1322679baf26c	no	36
Warnings:					
Information:					
3	Information Disclosure Statement Letter	IDS-5-TPO004C1.pdf	56321 494ddea72602585e09b3ec9e7c70861f785cba21	no	2
Warnings:					
Information:					
4	Information Disclosure Statement (IDS) Filed (SB/08)	1449-5-TPO004C1.pdf	55888 8dd0434df57aeaa6a2f4773f60d95916b9d3be06	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
5	NPL Documents	ChineseOfficeAction-TPO004C1.pdf	577884 a243ee21afd09a7477edd47fbf63fa7d7e8dafe3	no	9
Warnings:					
Information:					
		Total Files Size (in bytes):	3572264		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 798 932 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
01.10.1997 Bulletin 1997/40

(51) Int. Cl.[6]: **H04N 7/50**

(21) Application number: **97104977.0**

(22) Date of filing: **24.03.1997**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **29.03.1996 US 626184**

(71) Applicant:
MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.
Kadoma-shi, Osaka 571 (JP)

(72) Inventors:
- **Naimpally, Saipraisad V.**
  Langhorne, PA 19047 (US)
- **O'Hara, Joseph P.**
  Medford, NJ 08055 (US)
- **Meyer, Edwin Robert**
  Bensalem, PA 19020 (US)
- **Ryan, Robert T.**
  Langhorne,PA 19047 (US)

(74) Representative: **Schwabe - Sandmair - Marx**
Stuntzstrasse 16
81677 München (DE)

(54) MPEG-2 transport decoder

(57) A transport decoder 110, for receiving and processing a transport data stream using MPEG-2 formats, includes connections to a physical layer channel interface (channel interface) 112, a buffer memory 114, a host microprocessor 116, audio and video decoders 118/120, and clock signal circuitry 122. Upon receipt of a discontinuity indicator, the transport packet headers are parsed and transport packet payloads are stored in memory. Subsequently, upon receipt of a next program counter reference (PCR) value, a counter is loaded with the received program counter reference value. Next, the data stored in memory is searched for a time stamp and, when one is found, the time stamp is retrieved. Finally, a timer interrupt is set for a point in time approximately one frame time before the time stamp value, and, when the timer interrupt occurs, the program counter reference value is sent to the decoders, thereby providing sufficient time for the decoders to process data.

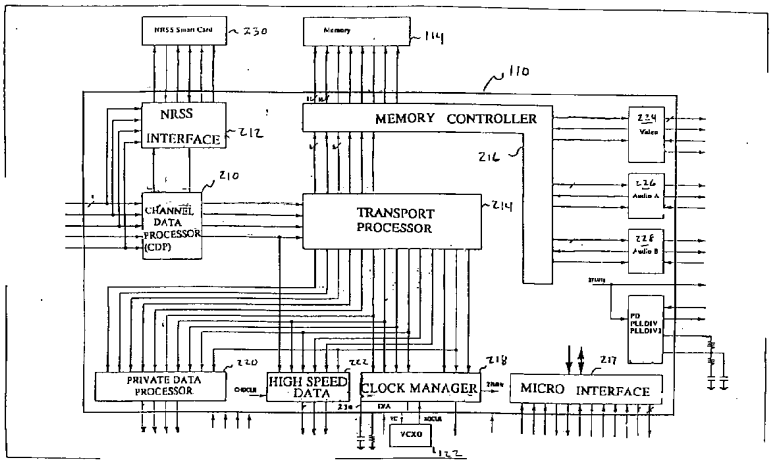


FIGURE 2A

EP 0 798 932 A2

Description

FIELD OF THE INVENTION

The present invention relates generally to data reception, processing and transmission of data encoded according to MPEG-2 standards and, more particularly, the present invention relates to the operation and interfacing of a transport decoder which handles an MPEG-2 format datastream.

BACKGROUND OF THE INVENTION

High Definition Television (HDTV) continues to make progress in its attempts to replace conventional television. Paving the way for this progress are various companies and associations working on standards to provide for a global market for HDTV.

One such group of companies is known as the "Digital HDTV Grand Alliance" including members such as AT&T, David Sarnoff Research Center, Massachusetts Institute of Technology and others. A comprehensive overview of the strides made by this group are presented in an article by Robert Hopkins entitled "Digital Terrestrial HDTV for North America: The Grand Alliance HDTV System" published in the IEEE Transactions on Consumer Electronics (Summer 1994). This article is herein incorporated by reference for all of its teachings regarding the background and basics of HDTV systems including the use of Program and Transport Packet Streams.

One standard that has been adopted by the Grand Alliance is the MPEG-2 standard for encoding video and audio information, developed by the Moving Pictures Expert Group (MPEG), a committee within the International Standards Organization (ISO). Accepted standards are periodically published such as, for example, the Video Section of Information Technology - Generic Coding of Moving Pictures and Associated Audio ISO/IEC 13818-2 (1995) (hereinafter "Video Section") and the Systems Section of Information Technology - Generic Coding of Moving Pictures and Associated Audio ISO/IEC 13818-1 (November 1994) (hereinafter "Systems Section") both of which are incorporated herein by reference for their teachings regarding established standards and formats.

The syntax for the MPEG-2 standard defines several layers of data records which are used to convey both audio and video data. To transmit information, a digital data stream, representing, for example, multiple video sequences, is divided into several smaller units and each of these units is encapsulated into a respective packetized elementary stream (PES) packet. For transmission, each PES packet is divided, in turn, among a plurality of fixed-length transport packets. Each transport packet contains data relating to only one PES packet. The transport packet also includes a header which holds control information, sometimes including an adaptation field, to be used in decoding the transport packet. However, PES data may include multiple elementary streams.

As set forth in the Systems Section, synchronization among multiple elementary streams is accomplished by periodically providing time stamps in the data stream. For instance, the decoding of N elementary streams is synchronized by adjusting the decoding of the streams to a common master time base rather than by adjusting the decoding of one stream to match that of another. The master time base may be the clock signal of one of the N decoders, or the clock signal of the data source, or it may be an externally supplied clock signal.

To aid in this synchronization process, included in the transport header of the MPEG-2 standard is a "discontinuity indicator". A syntax representation is provided in the Systems Section showing where in the transport header an exemplary "discontinuity indicator" resides. This syntax is re-presented below in Table I.

Table I -- Transport Stream adaptation field

Syntax	No.of Bits	Mnemonic
adaptation_field() {		
adaptation_field length	8	imsbf
if(adaptation_field_length > 0) {		
discontinuity_indicator	1	bslbf
random_access_indicator	1	bslbf
elementary_stream_priority_indicator	1	bslbf
PCR_flag	1	bslbf
OPCR_flag	1	bslbf
splicing_point_flag	1	bslbf
transport_private_data_flag	1	bslbf
adaptation_field_extension_flag	1	bslbf
if(PCR_flag == '1') {		
program_clock_reference_base	33	uimsbf
reserved	6	bslbf
program_clock_reference_extension	9	uimsbf
}		
if(OPCR_flag == '1') {		
original_program_clock_reference_base	33	uimsbf
reserved	6	bslbf
original_program_clock_reference_extension	9	uimsbf
}		
if(splicing_point_flag == '1') {		
splice_countdown	8	tcimsbf
}		
if(transport_private_data_flag == '1') {		
transport_private_data_length	8	uimsbf
for (i=0; i<transport_private_data_length;i++){		
private_data_byte	8	bslbf
}		
}		

Table I (continued)

Syntax	No.of Bits	Mnemonic
if(adaptation_field_extension_flag == '1'){		
adaptation_field_extension_length	8	uimsbf
ltw_flag	1	bslbf
piecewise_rate_flag	1	bslbf
seamless_splice_flag	1	bslbf
reserved	5	bslbf
if(ltw_flag == '1'){		
ltw_valid_flag	1	bslbf
ltw_offset	15	uimsbf
}		
if(piecewise_rate_flag == '1'){		
reserved	2	bslbf
piecewise_rate	22	uimsbf
}		
if(seamless_splice_flag == '1'){		
splice_type	4	bslbf
DTS_next_au[32..30]	3	bslbf
marker_bit	1	bslbf
DTS_next_au[29..15]	15	bslbf
marker_bit	1	bslbf
DTS_next_au[14..0]	15	bslbf
marker_bit	1	bslbf
}		
for (i=0;i<N;i++){		
reserved	8	bslbf
}		
}		
for (i=0);i<N;i++){		
stuffing_byte	8	bslbf
}		
}		
}		

As described in the Systems Section, the discontinuity indicator resides within an adaptation field within the Transport Header. The discontinuity indicator is a 1 bit field which, when set to a logical '1', indicates that the discontinuity state is true for the current Transport Stream packet. When the discontinuity indicator is set to a logical '0' or is not present, the discontinuity state is false. The discontinuity indicator is used to indicate two types of discontinuities, system time-base discontinuities and continuity counter discontinuities.

A system time-base discontinuity is indicated by the use of the discontinuity indicator in Transport Stream program identification (PID) packets designated as a Program Clock Reference (PCR) PID (refer to Systems Section 2.4.4.9 on page 50). Subsequently, a PCR value (i.e., reference to a master clock signal) is received.

The discontinuity indicator bit is set, for example, when program material which was recorded with one time base is inserted into a data stream that was recorded with another time base. The difference in time base appears as an unexpected PCR value.

When the discontinuity state is true for a Transport Stream packet of a PID designated as a PCR PID, the next PCR in a Transport Stream packet with that same PID represents a sample of a new system time clock for the associated program. The system time-base discontinuity point is defined to be the instant in time when the first byte of a packet containing a PCR of a new system time-base arrives at the input of the System Target Decoder (STD).

It should be noted that an STD is well known in the art and, briefly, represents a hypothetical decoder which is conceptually connected to an output of a transport encoder. It is required that a datastream that conforms to the MPEG-2 standard shall not cause dataflow problems for the STD (including buffer overflow). Additional discussion of the STD is

provided in the Systems Section.

Continuing, the discontinuity indicator bit may also be set to '1' in Transport Stream packets of the same PCR PID prior to the packet which contains the new system time-base PCR. In this case, once the discontinuity indicator has been set to '1', it continues to be set to '1' in all Transport Stream packets of the same PCR PID up to and including the Transport Stream packet which contains the first PCR of the new system time-base. After the occurrence of a system time-base discontinuity, no fewer than two PCRs for the new system time-base will be received before another system time-base discontinuity can occur. Further, except when a trick mode status is true, data from no more than two system time-bases shall be present in the STD buffers for one program at any time.

A potential problem occurs when a time-base discontinuity requires that multiple System Time Constant (STC) values (e.g., in a Clock Manager of a Transport Decoder and in external video and audio decoders) be updated, particularly when a substantial latency may exist between data that is presently being decoded by external decoders and data that is being received and stored in the buffers of a transport decoder. That is, typically a transport decoder is coupled to external video and audio decoders each of which maintain an STC or equivalent thereof. Thus, when a change in system time-base occurs, adjusting all of the STC's at the same time may create system synchronization problems.

With this in mind, an important concern, addressed by the present invention, is how a transport decoder can efficiently handle the occurrence of a system time-base discontinuity between buffers and decoders.

SUMMARY OF THE INVENTION

The present invention involves a system designed for receiving a transport data stream which includes transport packets having headers and payloads, the headers include a discontinuity indicator. The system including at least one buffer for temporarily storing the received transport data stream and at least one external decoder for decoding data output from the buffer for presentation purposes. The present invention is designed to handle the receipt of a discontinuity indicator by, upon receiving a discontinuity indicator, parsing the transport packet header and storing the transport packet payloads in memory. Upon receiving a next program counter reference value, a counter is loaded with the received program counter reference value. The data stored is then searched for a time stamp and, when one is found, the time stamp is retrieved. Finally, a timer interrupt is set for a point in time based on the retrieved time stamp, and, when the timer interrupt occurs, the program counter reference value is sent to the decoder, providing the new time stamp value insufficient time for the decoder to process data having the new time base without interfering with the processing of data having the previous time base.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is best understood from the following detailed description when read in connection with the accompanying drawing, in which:

Figure 1A shows a high-level functional block diagram of an exemplary digital transport decoder and its various interfaces.

Figure 1B shows a high-level flowchart illustrating the steps followed by one aspect of the present invention using the exemplary implementation shown in Figure 1A.

Figure 2A shows a high-level functional block diagram of an exemplary implementation of a the transport decoder shown in Figure 1A.

Figure 2B shows a high-level data/control flow block diagram of an exemplary implementation of the transport decoder shown in Figure 1A.

Figure 3 shows a functional block diagram of the memory controller used in the transport decoder shown in Figure 2.

Figure 4 shows an exemplary flowchart illustrating how the memory controller handles Pointers FIFO Entries.

Figure 5A shows an exemplary flowchart illustrating steps executed during the video decoder interface aspect of the present invention.

Figure 5B shows an exemplary flowchart illustrating steps executed during the audio decoder interface aspect of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Overview

As mentioned above, one important aspect of the present invention is how a transport decoder can efficiently handle a discontinuity indicator. Another aspect of the present invention is interfacing a transport decoder to a video decoder which can only handle elementary stream data. Still another aspect of the present invention involves interfacing a transport decoder to a bit rate-constrained audio decoder. To encompass the above listed aspects, the present invention includes a specially-designed MPEG-2 Transport Decoder.

To provide some context for the discontinuity aspect of the present invention, a real-world example of a time-base discontinuity would occur if a program were being broadcast, using MPEG-2 formats, via satellite to many receiving stations across the country. But, during the program, each station inserted its own respective commercials. In this situation, a discontinuity indicator would be set to indicate a gap or break in the time-base of the program in order to allow for the insertion of commercials, each having its own local time base. The program transmission from the satellite, however, would continue while the individual stations inserted their respective programs. Accordingly, the internal timing of the transport decoder for receiving the program broadcast from the local station with the inserted commercials would desirably track the appropriate time-bases when a discontinuity indicator is encountered.

Turning to the figures, Figure 1A shows a high-level functional block diagram of an exemplary digital transport decoder 110 and its various interfaces. As shown in Figure 1A, transport decoder 110 includes connections to a physical layer channel interface (channel interface) 112, buffer memory 114, a host microprocessor 116, external video and audio decoders 118 and 120, and clock signal circuitry (e.g., VCXO) 122. In the exemplary embodiment of the present invention, the transport decoder 110 supports one video interface and up to two audio interfaces (e.g., audio A and audio B).

Host microprocessor 116 is also coupled, external to transport decoder 110, to the video and audio decoders 118 and 120 by way of microbus 124. This connection may use parallel or serial data paths and depends on the requirements of the individual external decoders selected for use. In any event, as will be appreciated by those of ordinary skill in the art, this method of coupling the host microprocessor 116 to external decoders is well known.

Generally, the transport decoder 110 and each of the external video and audio decoders 118 and 120 include a System Time Constant (STC) value (not shown) maintained in a register or counter for the purpose of synchronizing the decoding and display of the received data stream.

As mentioned, a problem can occur when a time-base discontinuity occurs within the data stream, and the discontinuity requires STC values to be updated both in the transport decoder 110 and in the video and audio decoders 118 and 120. This is particularly a problem when a substantial latency may exist between data presently being decoded by the external decoders and data being received and stored in the transport processor 110.

In the present invention, microprocessor 116 has an important function in controlling the synchronization of the datastream and the operation of the video and audio decoders 118 and 120. In the present invention, microprocessor 116 has direct, memory-mapped access to various on-chip registers within the transport decoder 110. Microprocessor 116 also has access to buffer areas in the external memory 114 (through read, write and watermark pointers) and to on-chip 32-byte Read and Write FIFOs (shown in Figure 3). Microprocessor 116 is coupled to several internal functional blocks by an internal microbus 223 (shown in Figure 2B) and allows for various event and error conditions to be signaled via maskable interrupts.

Generally, as illustrated in the flowchart of Figure 1B, one aspect of the present invention, by way of the specially-designed transport decoder 110, performs the following steps after receiving of a discontinuity indicator, step 150:

1) because a discontinuity indicator is not necessarily in the same data packet as a next Program Clock Reference (PCR) value, when the next PCR value is received, a counter within Clock Management circuitry (shown in Figure 2A) contained within transport decoder 110 is loaded with the next PCR value, step 152;

3) subsequent packets, parsed and stored in memory, are searched for a time stamp and, when one is found, the time stamp value is retrieved, step 154;

4) a timer interrupt is set by the microprocessor 116 for a point in time that is just less than one frame time before the retrieved time stamp value, step 156; and

5) when this timer interrupt occurs, the current STC value is sent to the external video and audio decoders 118 and 120, step 158, thereby giving the external decoders sufficient time to process the data before the time stamp value in the datastream is compared to the counter value maintained in the video or audio decoders 118 or 120.

The implementation of these high level steps is described in detail below.

Figures 2A and 2B show high-level functional block diagrams of an exemplary implementation of the transport decoder 110 shown in Figure 1A. With reference to Figures 2A and 2B, the overall data flow in the transport decoder 110 is described.

As shown in the figures, the channel interface 112 of Figure 1A leads to a channel data processor (CDP) 210. Channel interface 112 also provides data to an NRSS interface 212 for purposes of decryption by the NRSS Smart Card 230. The EIA Standard for Conditional Access, Version 2.6, NRSS Committee (4/95) is hereby incorporated by reference for its teaching regarding the NRSS Smart Card.

The CDP 210 is provided to preprocess (e.g., detect, synchronize and validate) the packets in the received datastream. The CDP 210, after performing its processing, forwards its output data to transport processor (TPROC) 214 for further processing. TPROC 214 performs various processing, such as parsing, in accordance with a PID Table described below in detail.

The TPROC 214 interacts with several of the functional blocks within transport decoder 110 including a Memory Controller 216, a Clock Manager 218, a Private Data Processor 220, and a High Speed Data Port 222. As mentioned, the microprocessor 116 has direct access to various on-chip registers. This access is gained by way of a Micro Interface 217 and the internal microbus 223 (shown in Figure 2B). Memory Controller 216, in addition to being coupled to TPROC 214, is also coupled to external memory 114 and video and audio decoders 224, 226 and 228.

The details of the processing of the digital datastream, with respect to the individual functional blocks, is now described.

Channel Data Processor (CDP) 210

It is noted that the datastream from the channel interface 112 is passed to the CDP 210, after possible buffering (not shown).

The CDP 210 performs various preprocessing steps on the received datastream. In particular, CDP 210 determines the byte and frame boundaries in the datastream and converts the received datastream into an 8-bit parallel format. In the exemplary embodiment of the present invention, operations occurring in the transport decoder 110 subsequent to the CDP 210 are performed using a byte format.

To determine byte and frame boundaries, the CDP 210 synchronizes to a framing pattern contained in the header of a Transport Packet. In particular, this synchronization is achieved by searching for the MPEG-2 Transport Stream sync_byte (e.g., 47H) and verifying its occurrence at the start of a programmable number of consecutive transport packets.

In doing so, the CDP 210 checks for the sync_byte, "locks" on to the datastream and continues validating packets. When the sync_byte search is in progress, a Search_State bit is set in a CDP Status Register (not shown). When the sync_byte has been found and verification of successive occurrences is in progress, a Sync_State bit is set. Finally, a Lock_State bit is set when a predetermined search criterion is met. In the exemplary embodiment of the present invention, the search criterion is three consecutive validated packets. Also, a Sync_Hysterisis value, indicating the number of packets with corrupted sync_bytes that the CDP 210 is programmed to allow to pass through before declaring loss of Lock_State, is established. When this hysterisis value is exceeded, a loss of lock is declared. The CDP 210 then issues a Lock_Lost interrupt and transitions from the Lock_State back to the Search_State.

Additionally, if the channel interface 112 (shown in Figure 1A) provides a CHPSTRT signal, the transport decoder 110 may be configured to use this signal as a means to synchronize to the datastream by setting a Framing_Mode bit in a CDP Framing and Sync Control Register to logical "1". In this case, synchronization is achieved within one transport packet time after the CHPSTRT is asserted.

Alternatively, if an Ignore_Sync bit, in the CDP Framing and Sync Control Register, is set to logical "0", the CDP 210 verifies sync_byte integrity at the expected (first) byte location relative to the CHPSTRT signal transition and declares a lock condition if verification falls. If Ignore_Sync is set to 1, the CDP 210 does not verify sync integrity.

After achieving the Lock_State, the output of the CDP 210 can be passed to and processed by the TPROC 214, which further processes (e.g., filters, parses, and formats) the data now in byte format.

Transport Processor (TPROC) 214

Signals produced by the TPROC 214 control the flow of data to the Memory Controller 216, the Micro Interface 217, the Clock Manager 218, the Private Data Processor 220 and the High Speed Data Port 222.

In the exemplary embodiment of the present invention, the TPROC 214 processes the data stream under control of a PID Table. As described in the Systems Section, a PID is a 13-bit field in a Transport Stream Header, indicating the type of data stored in the packet payload. Some PID values are assigned and some are reserved. Additional details of the PID Table are described in Section 3.3 of the Systems Section of the above-referenced MPEG-2 specification.

Regarding the PID Table, the TPROC 214 is capable of concurrently processing up to 32 user-selectable PIDs. The PIDs which the transport decoder 110 is to process are specified in the PID Table. Bit fields in the PID Table specify

various processing options for the data.

In operation, the PID is extracted by the TPROC 214 from the header of an incoming transport packet and simultaneously compared to all of the entries in the PID Table. If a match is found, the data in the packet is processed according to the options set in the matching PID Table entry. If no PID Table entry matches, the packet is discarded.

Each location in the PID Table for which the Payload Format bits are not set to Discard (000), is associated with a channel number. In conjunction with the TPROC 214, the Memory Controller 216 is responsible for transferring data to a specific buffer area (channel) in RAM. Operation of the Memory Controller 216 is discussed in detail below with reference to Figures 2, 3 and 4.

In the exemplary embodiment of the present invention, the PID Table, maintained in TRPROC 214, is set up under software control by the microprocessor 116 via the internal microbus 223 during an initialization process.

Regarding PID Table entry format, a PID Table entry includes, among other information, 1) a 13-bit PID and 2) a 1-bit PCR PID field which, when set, indicates that this PID carries the PCR for the program being decoded. An exemplary PID Table entry also includes a 1 bit PES HDR bit which, when set, enables the RAM buffer address corresponding to a "Stream ID" in the PES header to be stored to a Pointers FIFO. An interrupt is issued by the Memory Controller 216 when it writes data to the Pointers FIFO. This interrupt is referred to as a DMA_MARK signal which serves to alert the microprocessor 116 of the delivery of the Stream_ID byte. In the exemplary embodiment of the present invention, Channel Numbers are assigned to PID Table entries in numerical order based on their location in the table, for example, the first entry is assigned DMA Channel 0, the next one DMA Channel 1 and so on.

The TPROC 214 indicates that transport packets with PCR PID should be processed by the Clock Manager 218 by placing a PCR_PID signal in the PID Table entry. The Clock Manager 218 then monitors the adaptation fields of such packets for a PCR_flag and extracts the PCR.

After separating the packets according to their PIDs, the TPROC 214 removes the headers of the transport packets and, in conjunction with the Memory Controller 216, stores packet payloads (e.g., portions of PES packets) for the respective PID's in sequential memory locations of the designated channel in the external memory 114. The data in the memory 114 represents PES packets.

The TPROC 214 also handles write operations to the external memory from the host microprocessor 116.

MEMORY CONTROLLER 216

The Memory Controller 216 stores the parsed payload of Transport Packets in external memory 114 and provides video, audio, PSI and other data (e.g., private data) upon demand to the Video Processor 224, up to two Audio Processors 226 and 228 and to the host microprocessor 116 (or microcontroller).

Figure 3 shows an exemplary functional block diagram of a Memory Controller suitable for use in the transport decoder 110 shown in Figures 2A and 2b. As shown in Figure 3, the Memory Controller 216 receives data from the TPROC 214 via a Transport In FIFO 310. Memory Controller 216 receives/transmits data from the microprocessor 116 via Micro Interface 217 including Micro-in FIFO 314 and Micro-out FIFO 316. It is noted that the Micro Interface 217, shown external to Memory Controller 216 in Figures 2A and 2B, is shown in Figure 3 for the sake of clearly illustrating its interrelationship with FIFOs 314 and 316.

The Memory Controller 216 also includes a Flow Control Unit 318 containing On Chip Memory, a DMA Controller 320, Datapath logic 322, and FIFOs 324, 326 and 328 for delivering data to the video and audio decoders 224, 226 and 228, respectively.

The Memory Controller 216 writes data into external memory from two sources: one is the TPROC 214 (through the Transport In FIFO 310) and the other is the microprocessor 116 (through the Micro In FIFO 314).

The Memory Controller 216 reads data from external memory and supplies it to the following four destinations 1) Video Interface (through the Video FIFO 324), 2) Audio A Interface (through the Audio A FIFO 326), 3) Audio B Interface (through the Audio B FIFO 328), and 4) microcontroller 116 (through the Micro Out FIFO 316).

The external memory space is divided into separate, non-overlapping buffer areas, one for each DMA Channel in use. Up to 32 channels can be used. A channel number is associated with data written into the two source FIFOs or read out of the four destination FIFOs.

Data provided by the TPROC 214 and written into the Transport In FIFO 310 carries a channel number tag. The channel number for each of the other FIFOs is defined by the microcontroller 116 in a DMA Interface Register. A 128-bit DMA Channel Configuration Register (Channel Register) is set up by the microprocessor 116 for each channel in use. The Channel Register contains configuration information such as the buffer space defined by Begin and End Addresses, whether the buffer is configured as a FIFO (read and write pointers wrap around, sometimes referred to as a circular queue) or as a Queue (read and write pointers do not wrap around), etc.

If the Channel Register allows, the DMA Controller 320 can obtain a new buffer when the one it is currently using becomes full. This feature is normally used only for buffers which store data that is destined for the host microprocessor 116, such as PSI sections. The Begin and End Addresses of these new buffers are stored in a New Buffer FIFO. Only the microprocessor 116 can write new buffers to this FIFO. Used buffer pointers and other types of pointers are written

by the DMA Controller 320 to the Pointers FIFO. In the exemplary embodiment of the present invention, only the microprocessor 116 can read the Pointers FIFO. These pointers are used by the microprocessor 116 to manage reading of buffer data and to return used buffers back to the New Buffer FIFO. The Pointers FIFO Entries Flowchart, shown in Figure 4, shows the entries to the Pointers FIFO for all possible signal conditions, Channel Register configurations and buffer conditions.

Continuing with Figure 3, the DMA Controller 320 and Datapath logic 322 provide all interface signals to the external memory. The Flow Control Unit 318 directs the DMA Controller 320 to service a specific interface FIFO (one of six) by providing the appropriate control signals to it. These include:

1. DMA Controller Start

This signal commands the DMA Controller 320 to start servicing a specific FIFO using the commands such as DMA Controller Operation, DMA Input FIFO Select and DMA Output FIFO Select. At the same time, Flow Control Unit 318 reads out the 128-bit Channel Register and makes it available on the data bus of the Channel Register (internal memory). The DMA Controller 320 reads this 128-bit word into its internal register and the Flow Control Unit 318 is then free to do other read/write operations as required on the internal memory.

2. DMA Controller Operation (1 bit)

This bit signals a Write or Read operation to be performed by the DMA Controller on the interface FIFOs.

3. DMA Input FIFO Select (1 bit)

Signals which of the two input FIFOs 310 or 314 to service.

4. DMA Output FIFO Select (2 bits)

Signals which of the four output FIFOs 316m 324, 326 or 328 to service.

5. Channel Data In Request/Channel Data/Channel Number/DMA Controller Stop:

Usually, the DMA Controller 320 continues to service a FIFO until it cannot service the FIFO any longer due to some condition such as a full interface FIFO, or no data in external memory, etc. When this type of condition occurs, the DMA Controller 320 asserts the Channel Data In Request to the Flow Control Unit 318 and provides the Channel Register data and Channel Number data to the Flow Control Unit 318. The Flow Control Unit 318 updates the Channel Register memory with the Channel Data (updated information includes, for example, the Write and Read Pointers) and directs the DMA Controller 320 to service the next FIFO in the loop, unless a specific FIFO needs emergency action (because, for example, it may be approaching a full or empty status).

Occasionally, the Flow Control Unit 318 may need to issue a DMA Controller Stop command to the DMA Controller 320 in response to a need to service another FIFO such as the Transport In FIFO which may have a higher priority. This command directs the DMA Controller 320 to stop servicing the FIFO it is currently servicing and write the Channel data to the Flow Control Unit 318.

The Channel Register is updated by the DMA Controller 320 with a Read Pointer, a Write Pointer and other information at the point at which it stops servicing a FIFO. Since the Flow Control Unit 318 writes this to the internal Channel Register memory, the DMA Controller 320 can resume service to the interrupted FIFO from the point at which it was interrupted.

With reference to the DMA Controller's participation in the overall operation of the discontinuity state aspect of the present invention, when a next PES packet containing a Stream ID byte is written to the Memory Controller 216 by TPROC 214, the Flow Control Unit 318 causes the DMA_MARK signal to go high. The transition of the DMA_MARK signal triggers an interrupt at microprocessor 116 and the address of the PES packet header is stored in an input pointer FIFO to which the microprocessor has access. Since this PES header may have a time-stamp, the microprocessor 116, responsive to the flag raised by the discontinuity indicator, searches the stored PES headers for a time stamp and, when it finds one, retrieves the time stamp value. Based on the retrieved timestamp value, microprocessor 116 sets a timer interrupt.

Referring back to Figure 3, in the exemplary embodiment of the present invention, internal memory of the Flow Control Unit 318 includes four blocks of 64x32 bits, each of which are assigned for Channel Registers, New Buffer FIFO and Pointers FIFO.

In the exemplary embodiment of the invention, each Channel Register has 128 bits and there may be a maximum of 32 channels, so up to 128x32 bits can be assigned to Channel Registers, in multiples of 128 bits.

Words of 32-bits are written to and read from the New Buffer and Pointers FIFOs. At least 128x32 bits of memory (i.e., 128 words) are available for the New Buffer and Pointer FIFOs. Additionally, if less than 32 channels are in use, unused capacity from the Channel Register memory can be allocated to the New Buffer and Pointer FIFOs.

Following are examples of memory allocation for 32 DMA channels and 16 DMA channels.

1) Memory allocation for 32 used channels

New Buffer FIFO=64x32 bits
Pointer FIFO=64x32 bits
Channel Registers=128x32 bits

2) Memory allocation for 16 used channels

New Buffer FIFO=64x32 bits
Pointer FIFO=128x32 bits
Channel Registers=128x16 bits

CLOCK MANAGER 218

The Clock Manager 218, shown in Figures 2A and 2B, of the transport decoder 110 synchronizes a local 27 MHz signal (generated by clock signal circuitry 122) using successive PCR values. The value of the counter in the Clock Manager 218 is referred to as the System Time Constant (STC) value. The microprocessor 116, in the exemplary embodiment of the present invention, has access to the STC value in the counter via the internal microbus 223.

The counter in the Clock Manager 218 is periodically synchronized to the PCRs carried in the transport data stream. When the first PCR in a data stream is encountered, the PCR is "jam loaded" in the counter to become the initial STC value. As subsequent PCR values are received, they are compared to the STC value to adjust the frequency of the clock signal circuitry 122 (e.g., VCXO). In the present invention, this is accomplished using a Digital-to-Analog converter (DAC) described in detail below. Once phase error adjustments are made, generally, the new PCR value is loaded as the STC value.

Normally PCR values occur once per frame. Under the MPEG-2 standard, they must occur at least in every third frame (one-tenth of a second).

The external video and audio decoders 118 and 120 are typically synchronized to the transport decoder 110 by updating the STC value in the external decoders based on the STC value maintained in the Clock Manager 218. The video and audio decoders 118 and 120 use the stored value to determine when data is to be decoded/displayed by comparing it to timestamps in the received PES datastream. It is noted that, in the exemplary embodiment of the present invention, the transport decoder 110, video decoder 118 and audio decoder 120 all receive the same 27 MHz clock signal generated by the Clock Manager 218.

The transport decoder 110 contains hardware support for locally generating a clock signal that is locked to the system clock signal for a selected program. As mentioned above, PCR values received on the specified PCR_PID are made available on a serial interface for use by other devices.

The Clock Manager 218 includes a System Time Counter (STC) (not shown), a System Time Clock Register (not shown), a Program Clock Reference Register(not shown), a Current STC Register (not shown), an Alarm Clock Interrupt Register (not shown) and logic (not shown) to latch PCRs extracted from the headers that are associated with a selected PID. It also contains the serial PCR interface, and a DAC Register for control of an external system clock signal control loop.

In the exemplary embodiment of the present invention, the STC is a 42-bit counter having a 9-bit base and 33-bit extension.

The STC base divides the nominal 27 MHz System Clock input signal by 300. The resulting 90 KHz signal drives the STC extension. The STC can be read or written by software executing on the microprocessor 116 at any time. Optionally, the STC can be automatically loaded with a received PCR after a discontinuity indicator has been encountered.

As mentioned above, PCRs may be automatically extracted from the incoming bit stream by setting the PCR_PID bit in a PID Table entry. Handling of the received PCRs depends upon options set by software and whether a discontinuity state exists for the current transport packet when the PCR is received.

A discontinuity state is recognized when a transport packet which contains the PCR_PID is received with a logic 1 in the discontinuity_indicator bit in the Transport header. The discontinuity state exists until either a transport packet containing the PCR_PID is received with a logic 0 in the discontinuity_indicator, or until the next PCR is found in a PCR_PID packet.

If enabled, the Clock Manager 218 automatically loads the received PCR into the STC when a PCR is received

while a discontinuity state exists. At this instant, an interrupt is issued to the host microprocessor 116.

The Clock Manager 218 contains an Initialize bit which, when set to logic 1 by software, causes the next PCR received to be loaded into the STC regardless of whether a discontinuity state exists or not. The Initialize bit is subsequently cleared automatically by the Clock Manager 218. The Initialize bit is set only when the STC does not contain a valid PCR (i.e., when a new PID has been selected.)

If no discontinuity state exists or if the automatic STC loading function is disabled, a received PCR is stored in the PCR register and (optionally) an interrupt is issued to the host microprocessor 116. At the same time that the PCR is captured, the current state of the STC is stored in the STC Register. The stored PCR and STC values are used to compute an error signal to control the external system clock signal (XOCLK).

When a Transport Packet is received which has a PID equal to the PCR_PID and which contains a PCR, the PCR is latched in the PCR Register. The current contents of the STC Counter are also latched in the STC Register at that time, and an interrupt is issued to the host microprocessor 116. Latching both values eliminates any uncertainty due to varying latency in the interrupt service routine. In the exemplary embodiment of the present invention, both registers consist of a 9 bit base and 33-bit extension. The base part is modulo 300, whereas the extension is binary. Before the register contents can be used in any calculations, they are converted to a single binary count by multiplying the extension by 300 and adding the base part to the result.

The Current STC Register is a read-only register which allows the microcontroller 116 to obtain the current contents of the STC at any time. The microcontroller 116 can initialize the STC of the Video Decoder or Audio Decoder with this value by way of the external microbus 124.

As described above, transport decoder 110 contains a 10-bit digital-to-analog converter (DAC) 230 for use in the clock adjustment control loop within the Clock Manager 218. The DAC 230 is write-only, i.e., software keeps a copy of the last value loaded into the DAC in memory. In operation, the DAC 230 is used to generate the control voltage for the external resonant crystal voltage controlled oscillator, VCXO 122, which is source of the system clock frequency (27 MHz nominal). By comparing the latched PCR and STC values, software can determine whether the local clock signal requires adjustment. For example, software may take the difference between the PCR and STC and add this error term to the current contents of the input register of DAC 230.

To review, the discontinuity aspect of the present invention includes:

1) The Channel Data Processor (CDP) 210 checks the synchronization byte in received transport packets to validate the packets. The Transport Processor (TPROC) 214 parses the transport header and passes the packet payloads to the proper location in the memory. Data is directed to different FIFO buffers (channels) in the memory based on the PID values in the transport headers.

2) Because a received discontinuity indicator is not necessarily in the same packet as the next PCR value, when the next PCR value is received, the TPROC 214 "jam loads" the counter in the Clock Manager 218.

3) When the next PES packet containing a Stream ID byte is written to the Memory Controller 216 by TPROC 214, the DMA_MARK signal goes high. This causes the Memory Controller 216 to interrupt the microprocessor 116 and store the address of the PES packet header in an input pointer FIFO. This PES header may have a time-stamp (either decode timestamp or presentation timestamp). The microprocessor 116, responsive to the interrupt from the Memory Controller 216, searches the PES headers for a time stamp and, when it finds one, retrieves the time stamp value.

4) The microprocessor 116 then sets a timer interrupt for a time that is just less than one frame time before the timestamp value.

5) When this timer interrupt occurs, the microprocessor 116 sends the new PCR value to the video and audio decoders by way of the external microbus 124. This gives the decoders sufficient time to process the data before its time-stamp is compared to the counter value in the video or audio decoder, without interfering with the processing of data that is reverenced to the prior PCR value.

VIDEO/AUDIO DECODERS 224, 226, 228

Once the transport packets are received, processed and stored, as mentioned above, Memory Controller 216 reads the PES packets out of memory 114 and forwards them to the external video and audio decoders. Before the PES packets reach the external decoders 118 and 120, however, they are processed by internal video and audio processors 224, 226 and 228.

An important role of these internal processors is to ensure that the stream of data leaving the transport decoder is compatible with the external decoders. That is, different external decoders have different data requirements.

For example, some currently available external video decoders can accept a stream of PES packets, including header and payload, essentially in the same form in which it is read out of memory. On the other hand, some external decoders can only accept for processing an elementary stream (i.e., PES packet payloads but no headers). Additionally, the latter type of decoder may require, in addition to the datastream, associated control information such as timestamps and byte counts. One such video decoder is the CL9100 made by C-Cube, the specification of which -- CL9100 Multimode Video Decoder User's Manual (10/94) -- is hereby incorporated by reference. Accordingly, video processor 224 is configurable, by way of microprocessor 116, to format the outgoing datastream to be compatible with the requirements of the selected external decoder.

With respect to handling the interface to an external video decoder such as a C-Cube CL9100, video processor 224 processes the stream of PES packets in order to strip off the PES header and extract, from the PES header, timestamp information. In addition, video processor 224 maintains a byte count, using a counter (not shown), for the PES packets forwarded to the external decoder. Thus, video processor 224 holds timestamp information, maintains a byte count and forwards the PES packet payloads on to the external video decoder. In order to get the timestamp and byte count information to the external decoder, video processor 224 sends an interrupt to microprocessor 116 which, when it responds, reads the timestamp and byte count information from the video processor 224 via the internal microbus 223 and delivers this control information to the external video decoder via the external microbus 124.

Sometimes, however, before microprocessor 116 is able to respond to the interrupt from video processor 224, another PES header with timestamp information arrives. This may occur, for example, if the microprocessor 116 was involved with a higher priority task for an extended period of time. In this case, video processor 224 stores the timestamp information along with a status. In particular, video processor 224 maintains a status register indicating whether microprocessor has "missed" any timestamps. When, in the exemplary embodiment of the invention, microprocessor 116 finally begins to read the timestamp information, if it has missed one or more timestamps as indicated in by the status, it only retrieves the most recent timestamp.

Additionally, since timestamp information and byte count information are to be provided to the external decoder, when the microprocessor 116 misses a timestamp, video processor 224 maintains a cumulative byte count for however many timestamps may arrive before microprocessor 116 reads the information. Thus, microprocessor 116 reads the most recent timestamp and cumulative byte count and delivers this information to the external video decoder. Subsequently, the status register and byte count may be reset.

As shown in the flowchart of Figure 5A, the headers of received data are stripped and parsed (step 510), timestamp information is stored and a byte count is maintained (step 512), an interrupt is issued by video processor 224 to microprocessor 116 alerting it that timestamp information is available for reading (step 514), if additional timestamp information is received before microprocessor reads the previous timestamp information, the new timestamp information is stored, the byte count continues to accumulate (step 516) and the video processor 224 issues another interrupt to the microprocessor 116, otherwise microprocessor reads the status register, the timestamp information and the byte count information and delivers it to the external video decoder (step 518).

Another example of accommodating particular external decoders involves the use of bit rate-constrained audio decoders such as the Zoran ZR38500 AC3 audio decoder, the Specification (10/94) for which is hereby incorporated by reference. Generally, most audio decoders present a data request indicating they can accept audio data and then the transport decoder 110, via the audio processors 226/228, supplies audio data until the external audio processor negates the requests. In other words, there exists a handshake mechanism.

Bit-rate constrained decoders, however, such as the Zoran AC3 audio decoder, require that audio AC3 data be delivered as audio data frames, each frame including a frame time and bit-rate in its header. Delivering data in excess of this predetermined bit rate can cause the internal buffers to overflow and, consequently, cause a loss of data. Furthermore, these bit-rate constrained decoders do not provide a Data-Ready signal to indicate that they are ready to accept additional audio data.

Thus, as with the video decoder 224, the audio processors 226 and 228, are configurable to accommodate the selected external audio decoder. It is noted that, in the exemplary embodiment of the present invention, audio processors 226 and 228, except for addresses which allow access by microprocessor 116, are essentially identical.

Accordingly, audio processors 226/228, when programmed to interface with this type of audio decoder, assumes that the decoder is always issuing a data request signal. However, as PES packet audio data is supplied to audio processors 226/228 from Memory Controller 216, the PES header of the received data is parsed by the audio processors. Contained in the parsed header is a frame size and bit rate for the audio data. From the frame size and bit rate, the audio processors 226/228 can determine the frame boundary (i.e., frame time). Having frame time information, audio processors 226/228 set a frame timer such that, when audio data is available, the timer is set, a frame of data is "burst" out to the external audio decoder then the audio processors 226/228 and then pause until the frame timer expires. When the timer expires, the cycle continues, one audio frame at a time, while additional frames are available for sending to the external audio decoder. Control information, such as timestamps, are retrieved by the microprocessor and delivered via the external microbus.

As shown in the flowchart of Figure 5B, PES packets are retrieved from memory (step 542), the headers of received

data are stripped and parsed (step 530), a frame time is determined (step 532), the timer begins running (step 534), audio processors 226/228 burst out a frame of data (step 536) and audio processors 226/228 pause until the timer expires (step 538), although, as shown, steps 542, 530 and 532 can be carried out during the pause. Then, the process can continue.

Although the invention is illustrated and described herein as embodied in a method and apparatus for updating the system time constant following a discontinuity in an MPEG-2 transport data stream, the invention is not intended to be limited to the details shown. Rather, various modifications may be made in the details within the scope and range of equivalents of the claims and without departing from the spirit of the invention.

Claims

1. A method for use in a transport decoder system designed for decoding a transport data stream which includes transport packets having headers and payloads, the headers including a discontinuity indicator, said system including at least one buffer for temporarily storing the received transport data stream and at least one decoder for decoding data output from the buffer for presentation purposes, each of said transport decoder and said decoder having separate counters, the method being a method for handling the receipt of a program counter reference (PCR) after a discontinuity state has been set comprising the steps of:

   parsing received transport packets and storing the data in memory;

   upon receiving a next PCR value, loading the transport decoder counter with the received program counter reference value;

   searching the data stored in memory which is received after the PCR value for a time stamp and retrieving the time stamp;

   setting a timer interrupt based on the retrieved time stamp value; and

   when the timer interrupt occurs, sending the PCR value to the counter of the decoder, thereby providing sufficient time for the decoder to process data.

2. The method of claim 1, wherein the timer interrupt is set for just less than one frame time before the time stamp value.

3. The method of claim 1, wherein the step of searching includes

   upon receipt of a transport packet which may contain a time stamp value, interrupting a host microcontroller to send the host microcontroller address information relating to the stored transport packet;

   subsequently, searching at the appropriate address location of the stored transport packet for a time stamp value.

4. The method of claim 1, further comprising the step of:

   before parsing, validating received transport packets by determining whether their boundaries correspond to predetermined packet boundaries and parsing only those packets which have been validated.

5. The method of claim 4, wherein the step of validating the received transport packet includes checking a synchronization byte in the received transport packet.

6. In a system designed for receiving a transport data stream which includes transport packets having headers and payloads, the headers including a discontinuity indicator, said system including at least one buffer for temporarily storing the received transport data stream and at least one decoder for decoding data output from the buffer for presentation purposes, each of said transport decoder and one decoder having separate counters, an apparatus for handling the receipt of a discontinuity indicator comprising the steps of:

   means for receiving and detecting a discontinuity indicator;

   means for parsing the transport packets and storing the data in memory;

means, upon receiving a next program counter reference value, for loading the transport decoder counter with the received program counter reference value;

means for searching the data stored in memory for a time stamp and, when one is found, for retrieving the time stamp;

means for setting a timer interrupt based on the time stamp value; and

means, when the timer interrupt occurs, for sending the program counter reference value to the counter of the one.

7. In a transport decoder system designed for decoding a transport datastream which includes transport packets having headers and payloads, the headers including a discontinuity indicator, said transport decoder coupled to at least one external decoder for decoding data output from the transport decoder, each of said transport decoder and said external decoders having separate counters for maintaining synchronization to the datastream, a method for handling the receipt of a discontinuity indicator comprising the steps of:

receiving the discontinuity indicator;

parsing the transport packet and storing the data in memory;

upon receiving a next program counter reference value, loading the transport decoder counter with the received program counter reference value;

searching the data stored in memory for a timestamp and, when one is found, retrieving the timestamp;

setting a timer interrupt for a point in time approximately one frame time before the time stamp value; and

when the timer interrupt occurs, sending the program counter reference value to the counter of the decoder, thereby providing sufficient time for the decoder to process data.

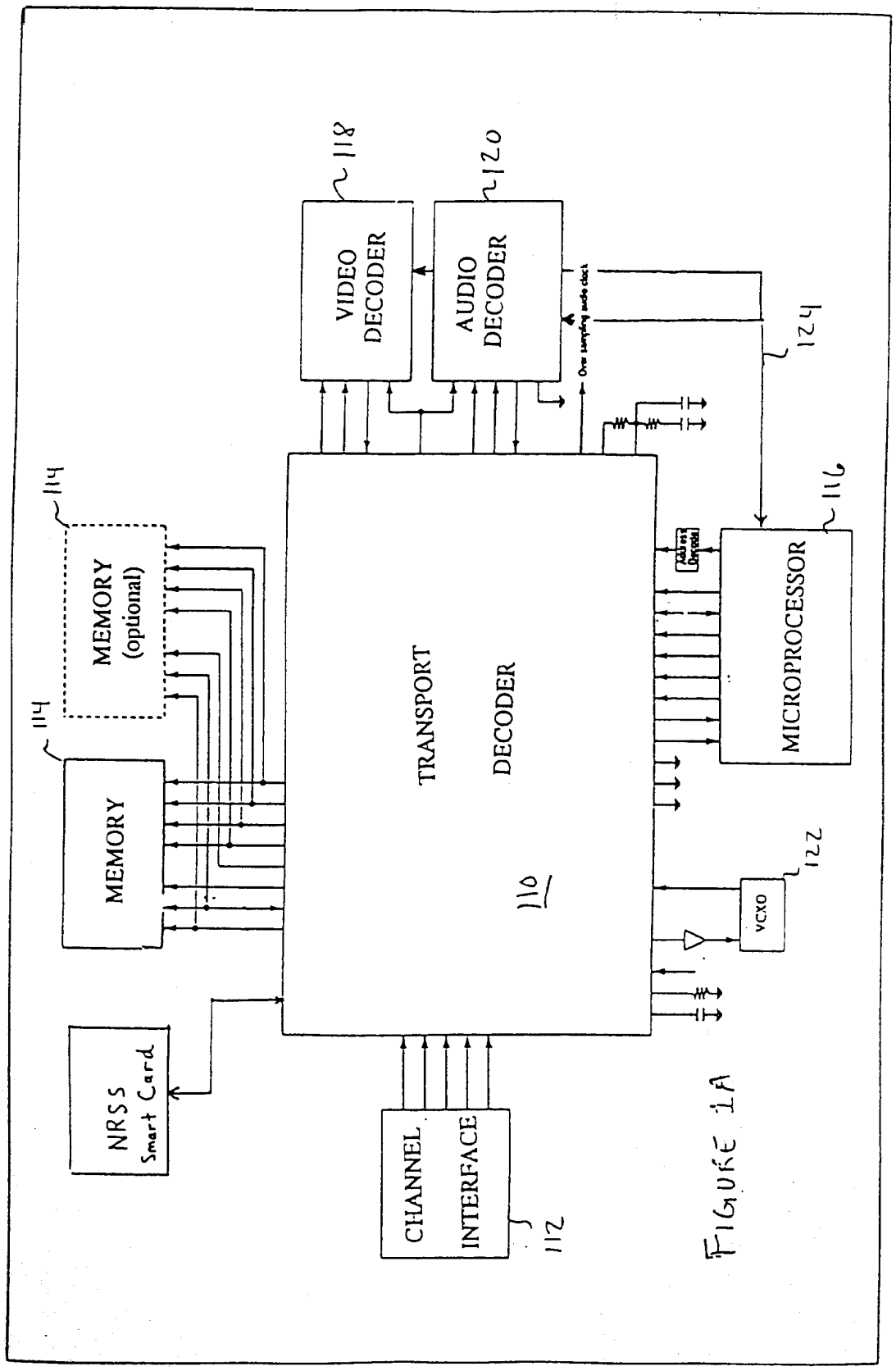

VIDEO DECODER
118
AUDIO DECODER
120
124
MEMORY (optional)
114
MEMORY
114
TRANSPORT DECODER
110
MICROPROCESSOR
116
VCXO
122
NRSS Smart Card
CHANNEL INTERFACE
112

FIGURE 1A

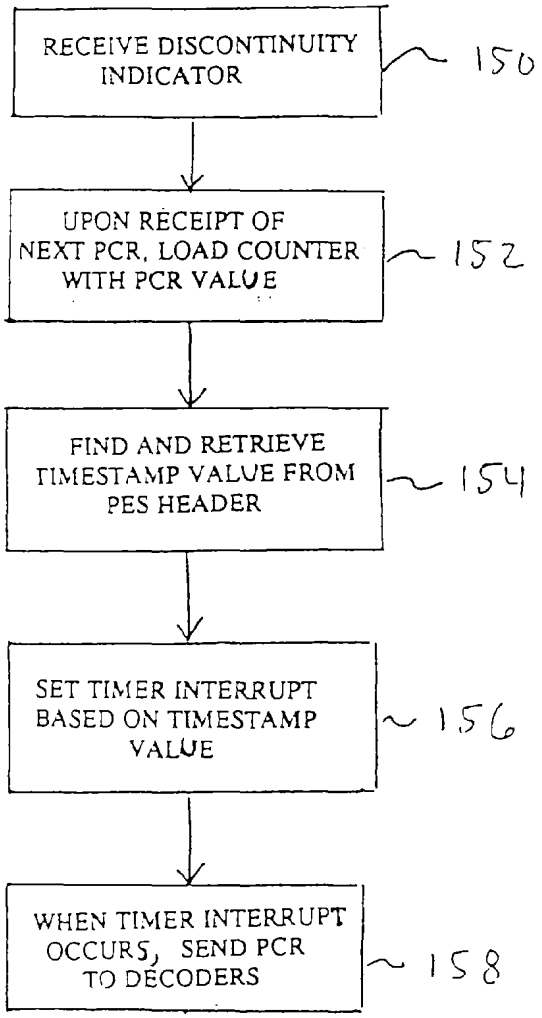

RECEIVE DISCONTINUITY INDICATOR
150
UPON RECEIPT OF NEXT PCR, LOAD COUNTER WITH PCR VALUE
152
FIND AND RETRIEVE TIMESTAMP VALUE FROM PES HEADER
154
SET TIMER INTERRUPT BASED ON TIMESTAMP VALUE
156
WHEN TIMER INTERRUPT OCCURS, SEND PCR TO DECODERS
158

FIG. 1B

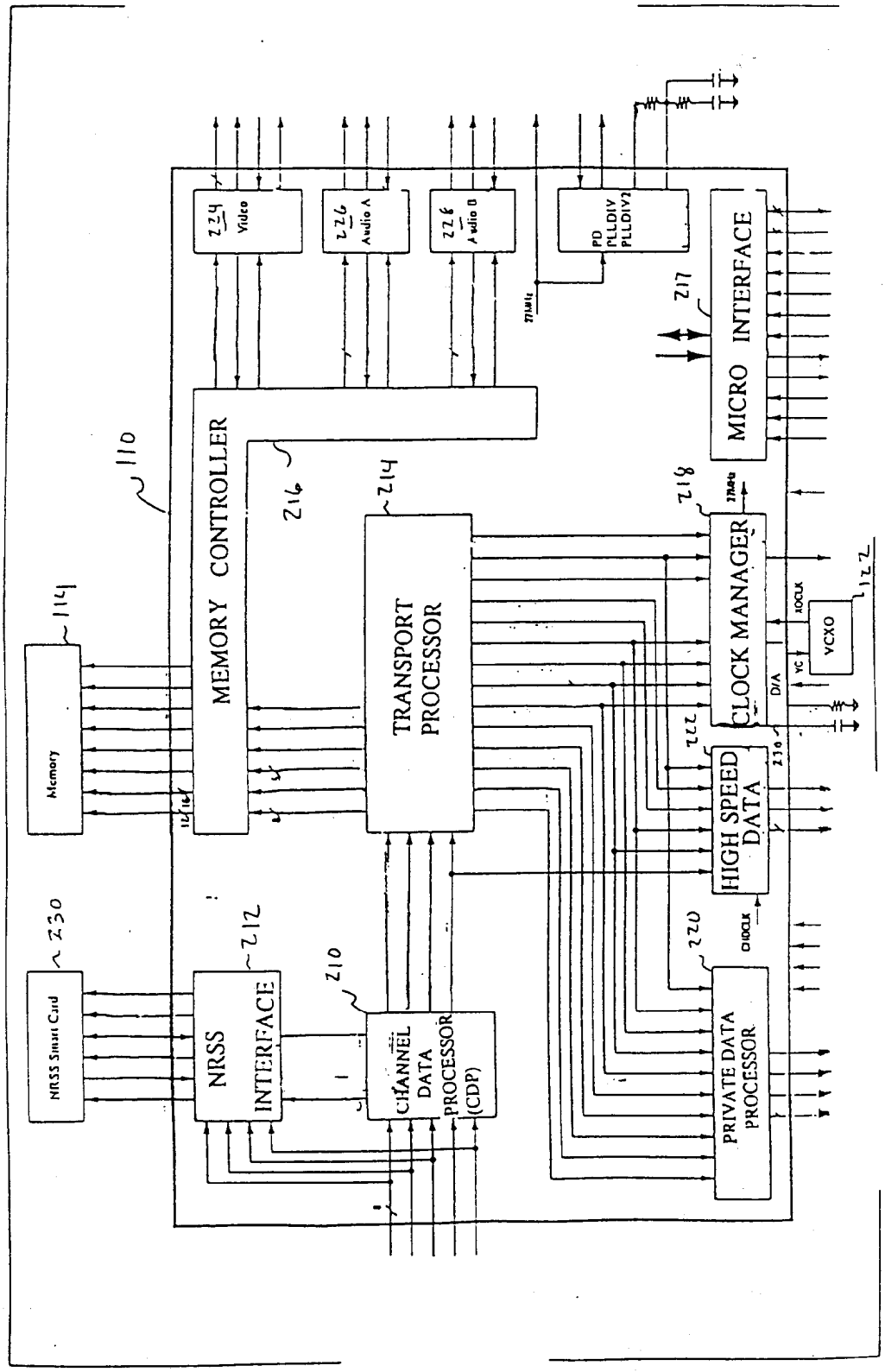

110
MEMORY CONTROLLER
216
TRANSPORT PROCESSOR
214
MICRO INTERFACE
217
CLOCK MANAGER
218
VCXO
222
HIGH SPEED DATA
224
PRIVATE DATA PROCESSOR
220
NRSS INTERFACE
212
CHANNEL DATA PROCESSOR (CDP)
210
Memory
114
NRSS Smart Card
230
Video
234
Audio A
226
Audio B
228
PD
PLLDIV
PLLDIV2

FIGURE 2A

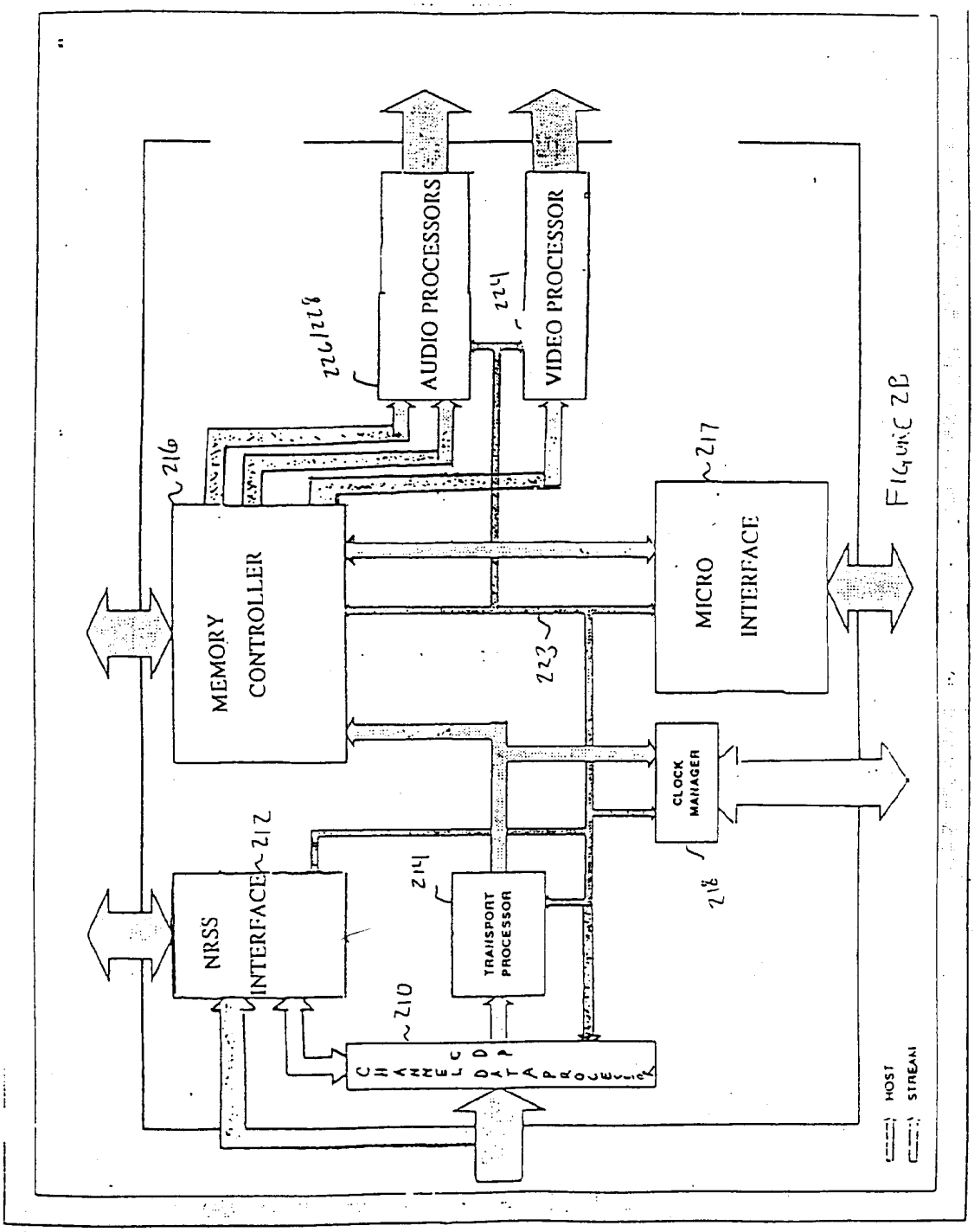

AUDIO PROCESSORS
VIDEO PROCESSOR
226/228
224
216
MEMORY CONTROLLER
217
MICRO INTERFACE
223
CLOCK MANAGER
218
212
NRSS INTERFACE
214
TRANSPORT PROCESSOR
210
CHANNEL DECODE DATA PROCESSOR
HOST
STREAM

FIGURE 2B

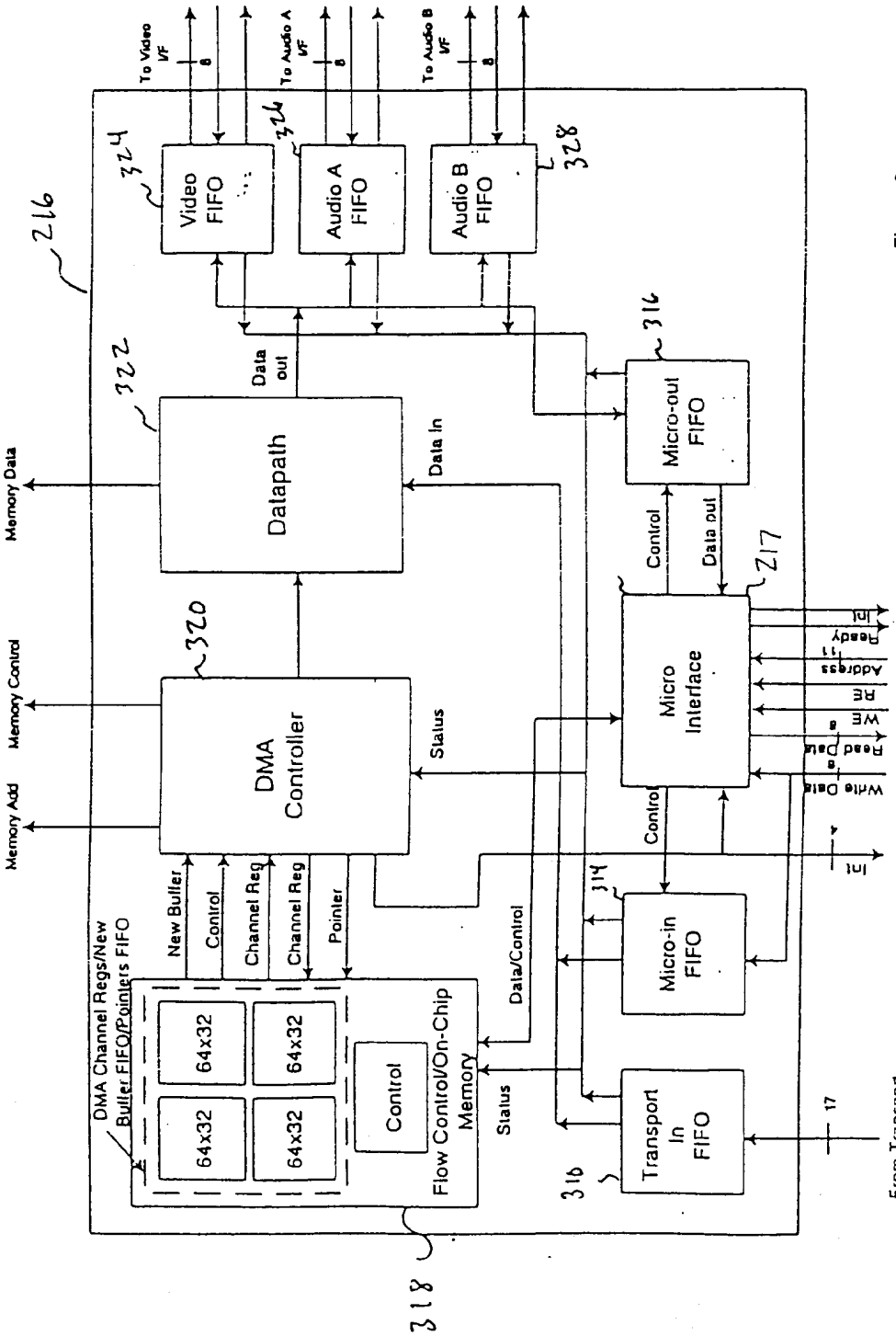
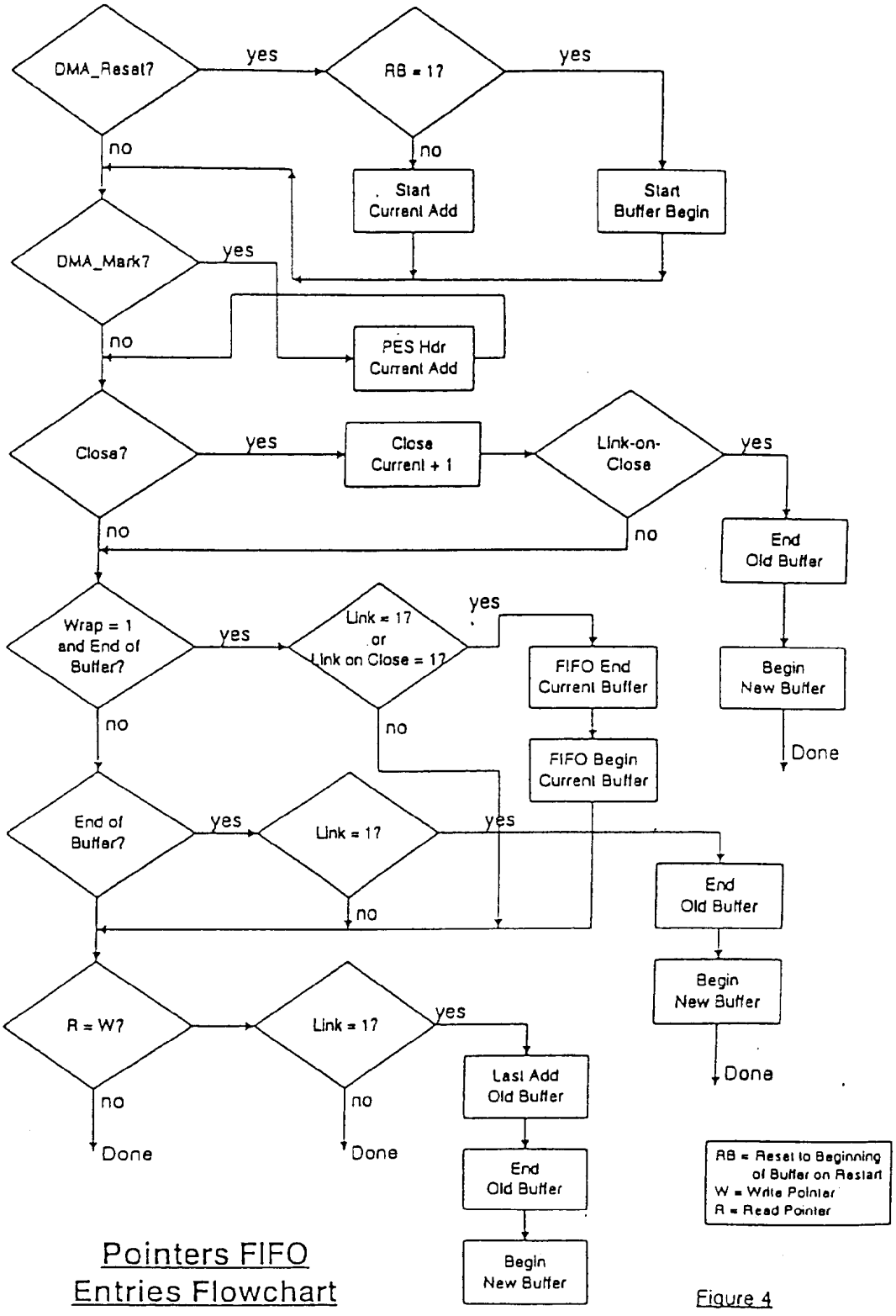


Figure 3

Memory Controller Block



Pointers FIFO Entries Flowchart

Figure 4

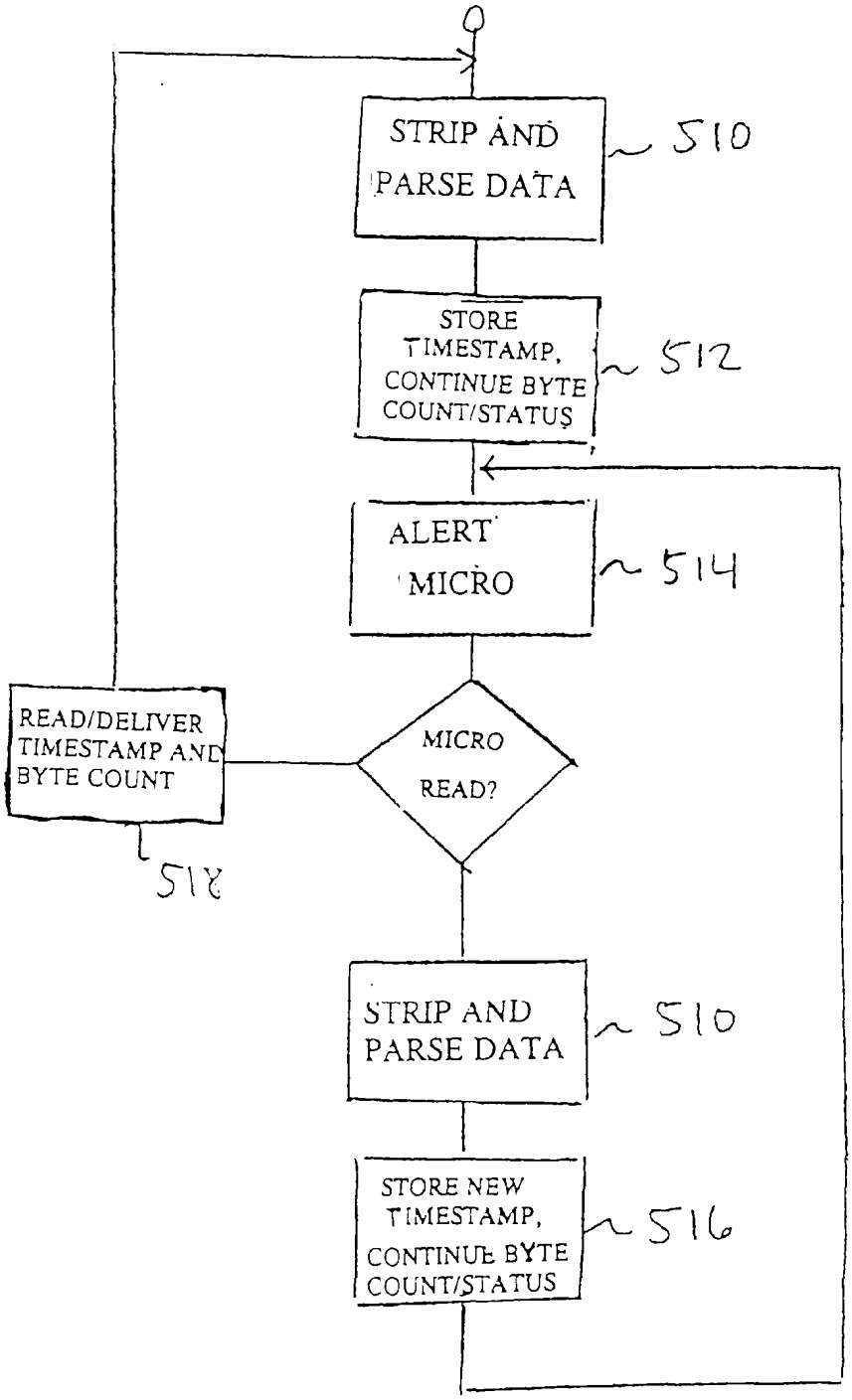

STRIP AND PARSE DATA
510
STORE TIMESTAMP, CONTINUE BYTE COUNT/STATUS
512
ALERT MICRO
514
MICRO READ?
READ/DELIVER TIMESTAMP AND BYTE COUNT
518
STRIP AND PARSE DATA
510
STORE NEW TIMESTAMP, CONTINUE BYTE COUNT/STATUS
516

FIGURE 5A

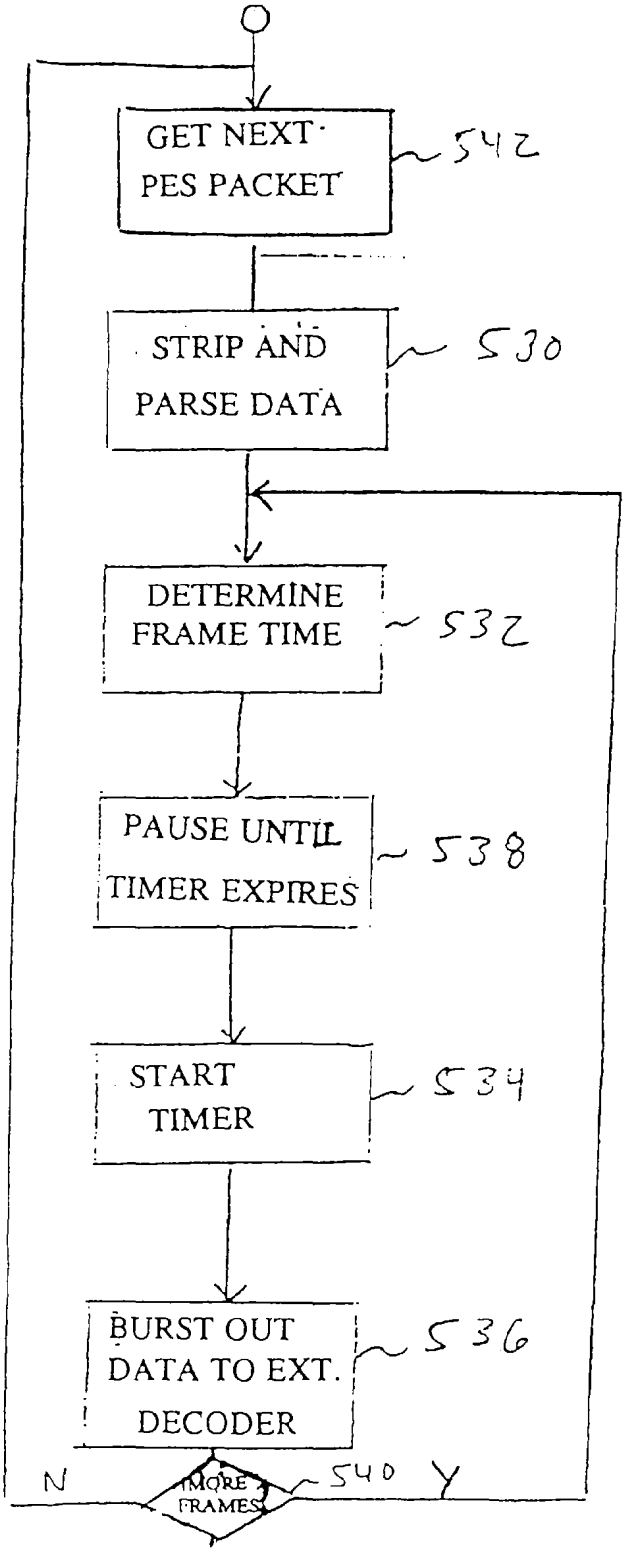

GET NEXT PES PACKET
542
STRIP AND PARSE DATA
530
DETERMINE FRAME TIME
532
PAUSE UNTIL TIMER EXPIRES
538
START TIMER
534
BURST OUT DATA TO EXT. DECODER
536
MORE FRAMES
540
N
Y

FIGURE 5B

[19]中华人民共和国专利局

[51]Int.Cl[6]
H04N 7/06

[12] 发明专利申请公开说明书

[21] 申请号 97103739.6

[43]公开日 1998 年 2 月 18 日

[11] 公开号 CN 1173783A

[22]申请日 97.3.31

[30]优先权

[32]96.3.29 [33]US[31]08 / 626,184

[71]申请人 松下电器产业株式会社

地址 日本大阪

[72]发明人 塞卜瑞舍·V·莱姆波利

约瑟夫·P·奥哈拉

爱德温·罗伯特·梅依尔

罗伯特·T·莱恩

[74]专利代理机构 永新专利商标代理有限公司

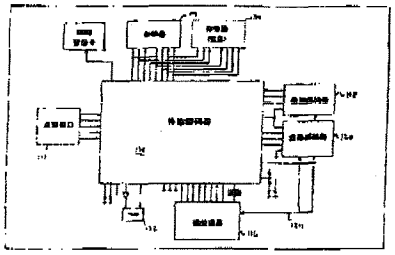
代理人 程 伟

权利要求书 3 页 说明书 24 页 附图页数 8 页

[54]发明名称 用于处理非连续程序数据的数字视频解码器

[57]摘要

接收和处理使用 MPEG-2 格式的传输数据流的传输解码器包括：与物理层通道接口的连接，缓冲存储器，主微处理器，音频和视频解码器，时钟信号电路。收到不连续性指示位时对传输包首部进行语法分析并将有效部分存于存储器。收到下一程序计数器参考（PCR）值时将其载入计数器。在存储器内搜索时间标记，若发出则将其恢复。在时间标记值前约一帧时刻设置定时器中断，该中断发生时，程序计数器参考值被送往解码器，为解码器提供足够数据处理时间。



(BJ)第 1456 号

权 利 要 求 书

1. 一种用于传输解码器系统的方法，所述系统用于对传输数据流解码，该数据流包括由首部和有效部分组成的传输包，首部包括不连续性指示位，所述系统包括至少一个缓冲区和至少一个解码器，前者用于临时存储接收到的传输数据流，后者用于对来自缓冲区的数据输出解码供显示之用，所述每个传输解码器和解码器都有各自的计数器，此方法用于在设定了不连续状态后处理程序计数器参考（PCR）的接收，其特征在于包括以下步骤：

对接收到的传输包进行语法分析，并将数据存入存储器；

一旦接收到下一个 PCR 值，将接收到的程序计数器参考值载入传输解码器计数器中；

对 PCR 值之后接收并储存在存储器中的数据进行数据标记搜索并将检索该数据标记值；

基于检索后的时间标记值设置定时器中断；以及

当定时器中断发生时，将 PCR 值传送给解码器的计数器，从而为解码器提供充足的时间处理数据。

2. 根据权利要求 1 所述的方法，其中，在时间标记值之前不足一帧的时间处设置定时器中断。

3. 根据权利要求 1 所述的方法，其中的搜索的步骤包括：

一旦接收到含有时间标记值的传输包，中断一台主微控制器以发出与存储的传输包相关的主微控制器地址信息；以及

接着，在存储的传输包的适当地址位置搜索时间标记值。

4. 根据权利要求 1 所述的方法，还包括以下步骤：

在进行语法分析之前，通过确定接收到的传输包的边界是否与预确定的包边界符合使它们生效，并且只对生效的包进行语法分析。

5. 根据权利要求 4 所述的方法，其中使接收到的传输包生效的步骤包括检查接收到传输包中的同步位。

6. 在用于接收传输数据流的系统中的一种处理不连续性指示位的接收的装置，该数据流包括由首部和有效部分组成的传输包，首部包括不连续性指示位，所述系统包括至少一个缓冲区和至少一个解码器，前者用于临时存储接收到的传输数据流，后者用于解码来自缓冲区的数据输出供显示之用，所述每个传输解码器和解码器都有分离的计数器，所述装置的特征在于包括：

接收和检测不连续性指示位装置；

对传输包进行语法分析并将数据存储在存储器中的装置；

一旦接收到下一个程序计数器参考值，将接收到的程序计数器参考值载入传输解码器计数器中的装置；

在存储器中存储的数据中搜索时间标记并在发现后检索时间标记值的装置；

基于时间标记值设置定时器中断的装置；以及

当定时器中断发生时，将程序计数器参考值发送到该计数器的装置。

7. 在一个用于解码传输数据流的传输解码器系统中，该数据流包括由首部和有效部分组成的传输包，首部包括不连续性指示位，所述传输解码器与至少一个外部解码器配合以解码来自传输解码器的数据输出，所述每个传输解码器和外部解码器都有分离的计数器以保持

与数据流同步，一种处理不连续性指示位的接收的方法包括以下步骤：

接收不连续性指示位；

对传输包进行语法分析并将数据存储在存储器中；

一旦接收到下一个程序计数器参考值，将接收到的程序计数器参考值载入传输解码器计数器中；

在存储器存储的数据中搜索时间标记并在发现后检索时间标记值；

在时间标记值前大概一帧的时间处设置定时器中断；以及

当定时器中断发生时，将程序计数器参考值发送到解码器的计数器中，从而为解码器提供充足的时间进行数据处理。

说 明 书

用于处理非连续程序数据的数字视频解码器

本发明涉及根据 MPEG-2 标准编码的数据进行数据接收、处理、及传输，具体来说，本发明涉及处理 MPEG-2 格式数据流的传输解码器的操作和接口。

高清晰度电视(HDTV)在试图替代传统的电视机方面继续取得进展。多家公司和协会正努力制定多种标准以供应 HDTV 的全球市场，从而为这种进展铺平道路。

其中有一个公司集团又被称为“数字高清晰度电视大联盟”，它包括 AT&T、David Sarnoff 研究中心、麻省工学院和其他一些机构。Robert Hopkins 在 1994 年夏季的《消费者电子 IEEE 交易》上发表了一篇以“北美数字陆地高清晰度电视：大联盟高清晰度电视系统”为名的文章，全面概述了该集团的长足进步。在此将引用该文章关于 HDTV 系统的背景和基础知识方面的教导以供参考，其中包括对程序和传输包流的使用。

大联盟采用由国际标准组织(ISO)下属的动态图像专家组(MPEG)开发的 MPEG-2 标准对视频和音频信息进行编码。被接受的标准定期发表在下列刊物上：动态图像和相关音频 ISO/IEC 13818 - 2(1995 年)的通用编码：信息工程的视频部分(下文简称为“视频部分”)，动态图像和相关音频 ISO/IEC 13818 - 1(1994 年 11 月)的信息工程的系统部分(下文简称为“系统部分”)。在此将引用二者关于已建立的标准和格式方面的教学资料以供参考。

MPEG-2 标准的语法定义了几种数据记录层，用来传送音频和视频数据。例如，为了传输信息，一个表示多视频序列的数字数据流被分为

多个较小的单元。每个单元被分别封装成多个打包的基本流(PES)包中。为了传输需要，每一个打包的基本流包被依次分割成多个定长传输包。每个传输包包含的数据只与一个打包的基本流包有关。传输包还包括一个首部，其中带有控制信息，有时也包括适应字段子段，用于对传输包解码。然而，打包的基本流数据也可能包括多种基本流。

正如在系统部分所述，多种基本流之间的同步是通过定期在数据流中提供时间标记实现的。例如，N 个基本流的解码是通过将流的解码校正到一个通用主时基的方式实现的，而不是调整一种流的解码以适应另一种流的解码。主时基可能是 N 个解码器中某一个的时钟信号，或者是数据源的时间信号，还可能是一个外部提供的时钟信号。

为了有助于该同步处理，MPEG-2 标准的传输首部中还包括一个不连续性指示位。系统部分提供了一种语法表示以显示一个典型的“不连续性指示位”在传输首部中的位置。该语法在以下的表 1 中有具体描述。

表 1

传输流适应字段

语法	位数	助记码
adaptation_field() {		
adaptation_field length	8	imsbf
if (adaptation_field_length > 0) {		
discontinuity_indicator	1	bslbf
random_access_indicator	1	bslbf
elementary_stream_priority_indicator	1	bslbf
PCR_flag	1	bslbf
OPCR_flag	1	bslbf
splicing_point_flag	1	bslbf
transport_private_data_flag	1	bslbf
adaptation_field_extension_flag	1	bslbf
if (PCR_flag == '1') {		
program_clock_reference_base	33	uimsbf

reserved	6	bslbf
program_clock_reference_extension	9	uimsbf
}		
if (OPCR_flag == '1') {		
original_program_clock_reference_base	33	uimsbf
reserved	6	bslbf
original_program_clock_reference_extension	9	uimsbf
}		
if (splicing_point_flag == '1') {		
splice_countdown	8	tcimsbf
}		
if (transport_private_data_flag == '1') {		
transport_private_data_length	8	uimsbf
for (i=0; i<transport_private_data_length; i++){		
private_data_byte	8	bslbf
}		
}		
if (adaptation_field_extension_flag == '1') {		
adaptation_field_extension_length	8	uimsbf
ltw_flag	1	bslbf
piecewise_rate_flag	1	bslbf
seamless_splice_flag	1	bslbf
reserved	5	bslbf
if(ltw_flag == '1'){		
ltw_valid_flag	1	bslbf
ltw_offset	15	uimsbf
}		
if(piecewise_rate_flag == '1'){		
reserved	2	bslbf
piecewise_rate	22	uimsbf
}		
if(seamless_splice_flag == '1'){		
splice_type	4	bslbf
DTS_next_au[32..30]	3	bslbf
marker_bit	1	bslbf
DTS_next_au[29..15]	15	bslbf

```
                    marker_bit                          1       bslbf
                    DTS_next_au[14..0]                  15      bslbf
                    marker_bit                          1       bslbf
                }
                for (i=0;i<N;i++){
                    reserved                            8       bslbf
                }
            }
            for (i=0);i<N;i++){
                stuffing_byte                           8       bslbf
            }
        }
    }
```

如系统部分中所述，不连续性指示位在传输首部内的适应字段中。不连续性指示位是一个 1 位的字段，设为逻辑值“ 1 ”时，表示当前的传输流包的不连续性状态为真。当不连续性指示位设为逻辑值“ 0 ”或不存在时，不连续性状态为假。不连续性指示位用于表示下列两种不连续性状态：系统时基不连续和连续性计数器不连续。

通过在被指定为程序时钟参考（ Program Clock Reference — PCR ）程序标识 的传输流程序标识（ PID ）中使用不连续性指示位，可以指示系统时基的不连续性（参见系统部分第 50 页的 2.4.4.9 ）。接下来，可以接收到一个程序时钟参考（ PCR ）值（即所谓主时钟信号）。

例如，当使用一个时基记录的节目材料被插入采用另一个时基记录的数据流中时，设定不连续性指示位。时基中的差别表现为一个不可知的程序时钟参考（ PCR ）值。

当一个被指定为 PCR 程序标识 的程序标识 的传输流包的不连续状态为真时，使用同一个程序标识 的传输流包中下一个 PCR 表示相关节目的新系统时间时钟取样。当包括一个新系统时基的 PCR 的包的第一个字

节到达系统目标解码器（ STD ）的输入端时，系统时基不连续点被定义为这个时刻。

值得注意的是：系统目标解码器在先有技术中广为人知，简而言之，它是表示一个在概念上与传输编码器的输出端相连的假定解码器。要求的条件是符合 MPEG-2 标准的数据流不会引起系统目标解码器的数据溢出问题（其中包括缓冲区溢出）。对系统目标解码器的其他讨论在系统部分给出。

紧接着，在传输流包中不连续指示位也可能被置为“ 1 ”。这些传输包都属于包含新的系统时基 PCR 的包的前一个程序时钟参考 程序标识。在这种情况下，一旦不连续性指示位被置为“ 1 ”，在同一个程序 PCR 标识的所有传输流包中它都被置“ 1 ”，该程序 PCR 标识符合和包括包含新系统时基的第一个 PCR 的传输流包。当系统时基的不连续性发生时，新的系统时基的不少于两个 PCR 在另一个系统时基不连续性发生之前被接收。进一步来说，除非欺骗模式状态为真，否则来自不多于两个系统时基中的数据会在任意时间出现在一个程序的系统目标解码器缓冲区中。

当时基不连续性要求更新多系统时间常数（ STC ）值时（即在传输解码器的时钟管理器和在外部的视频和音频解码器中），特别是当等待时间可能存在于正被外部解码器解码的数据和正被接收并储存在传输解码器的缓冲区中的数据之间时。换句话说，典型的情况是一个传输解码器与外部的视频和音频解码器耦合，后者中每一个都保存一个 STC 或等价的值。因此，当系统时基发生改变时，同时校正所有的 STC 值就可能引发系统的同步问题。

考虑到这个因素，本发明所要解决的一个重要问题在于传输解码器怎样才能有效地处理缓冲器和解码器之间的系统时基不连续性问题。

本发明包括一个用于接收传输数据流的系统，该数据流包括具有不连续性指示位的首部和有效内容的传输包。该系统至少包括一个缓冲存储器和一个外部解码器，前者用于临时储存接收到的传输数据流，后者用于解码来自缓冲区的数据输出以供显示之用。本发明用于处理对不连续性指示位的接收，其手段是通过接收到不连续性指示位后对传输包的首部进行语法分析，以及将传输包的有效内容存入存储器。一旦接收到下一个程序计数器参考值，一个计数器中就会装入接收到的程序计数器参考值。然后在储存的数据中搜寻时间标记，一旦发现，就检索该时间标记。最后，基于已检索到的时间标记为某一个时间点设置一个定时器中断，并且当定时器中断发生时，将程序计数器参考值送往解码器，同时向解码器提供新时间标记值不充分时间，使之能处理含有新的时基的数据，而无需干扰对含有前一时基的数据的处理。

图 1A 示出了一个典型的数字传输解码器及其各种接口的高级功能框图。

图 1B 示出了一个高级流程图，描述了采用图 1A 所示的实施例时本发明的一个方面采取的步骤。

图 2A 示出了图 1A 所示的传输解码器的典型实现的高级功能框图。

图 2B 示出了图 1A 所示的传输解码器的典型实现的高级数据/控制流程框图。

图 3 示出了图 2 所示的传输解码器中使用的存储器控制器的功能框图。

图 4 示出了用以描述存储器控制器如何处理指针 FIFO 项的典型流程图。

图 5A 示出了在本发明的视频解码器接口部分执行的步骤的典型流程图。

图 5B 示出了在本发明的音频解码器接口部分执行的步骤的典型流程图。

概述

如上文所述，本发明的一个重要方面在于解决传输解码器如何高效地处理不连续性指示位的问题。本发明的另一个重要方面在于处理传输解码器和仅能处理基本流数据的视频解码器之间的接口问题。另外，本发明还有一个方面涉及传输解码器和位速率限制音频解码器之间的接口问题。为了全面介绍以上的各个方面，本发明包括一个特别设计的 MPEG-2 传输解码器。

如果一个节目使用 MPEG-2 格式通过卫星向全国的多个接收站播放，一个时基不连续的实例就出现了。这就是为介绍本发明的不连续性部分提供的具体环境。但是，在节目播出的过程中，每个接收站会加入自己的商业广告节目。在这种情况下，应设置一个不连续性位来指示节目时基中的空白或间断，以便于自己拥有本地时基的商业广告的插入。

然而，当各个接收站插入各自的节目时，来自卫星的节目传输并不会中断。由此可知，当遇到不连续性指示位时，接收来自本地接收站并带有插入广告内容的节目的传输解码器的内部定时会跟踪适当的时基。

再看看附图，图 1A 示出了一个典型的数字传输解码器 110 的高级功能框图以及各种接口。如图 1A 所示，传输解码器 110 包括与以下各部分的连接：物理层通道接口（又叫通道接口） 112 、缓冲存储器 114 、主微处理器 116 、外部视频和音频解码器 118 和 120 以及时钟信号电路（例如 VCXO ） 122 。在本发明的典型实施例中，传输解码器 110 支持一个视频接口和多达两个音频接口（例如，音频接口 A 和 B ）。

处于传输解码器 110 外部的主微处理器 116 也通过微总线 24 与视频和音频解码器 118 和 120 耦合。这种连接可以使用并行或串行数据路径，并根据被选择使用的单个外部解码器的要求而变化。在任何情况下，将主微处理器 116 和外部解码器连接起来的方法都是众所周知的，同时也为拥有普通技艺的人员所理解。

一般来说，传输解码器 110 和每个外部视频和音频解码器 118 和 120 都包括一个的系统时间常数（ STC ）值（图中未给出）。该值储存在寄存器或计数器中用于使接收到的数据流的解码和显示同步。

如上所述，当时基不连续在数据流中发生时可能出现问题，而且这种不连续要求 STC 值在传输解码器 110 和视频及音频解码器 118 和 120 中同时更新。特别容易发生问题的时间是当等待时间可能存在于正被外部解码器解码的数据和正被接收并储存在传输处理器 110 中的数据之间时。

在本发明中，微处理器 116 在控制数据流的同步和视频及音频解码器 118 和 120 的运作方面起着十分重要的作用。在本发明中，微处理器 116 对传输解码器 110 中的多个芯片寄存器能进行直接的存储器映射式存取。微处理器 116 还能通过读、写、和水印指针对外部存储器 114 内的缓冲区和芯片上 32 位可读写 FIFO （如图 3 中所示）进行存取。微处理器 116 通过一个内部微总线 223（如图 2B 所示）与多个内部功能块连接，从而允许各种事件和错误状态通过可屏蔽中断用信号发出。

一般来说，如图 1B 所示的流程图中，本发明的一个方面通过特别设计的传输解码器 110 在接收到不连续性指示位（步骤 150 ）后执行以下步骤：

1) 由于不连续性指示位不必和下一个程序时钟参考值（ PCR)处于同一个数据包中，所以当下一个程序时钟参考 值被接收后，位于传输

解码器 110 中的时钟管理电路（如图 2A 中所示）内的计数器中将装载下一个程序时钟参考值（ PCR ）（步骤 152 ）；

3) 在经过语法分析并存储在存储器中的后续包中搜索时间标记，一旦找到某一个，则将检索时间标记值（步骤 154 ）；

4) 由微处理器 116 在某个时间点设置定时器中断，该时间点只比检索的时间标记值提前不到一帧的时间（步骤 156 ）；而且

5) 当定时器中断发生时，当前 STC 值被送往外部视频和音频解码器 118 和 120 （步骤 158 ），从而给外部解码器足够的时间以抢在数据流中的时间标记值与保存在视频或音频解码器 118 或 120 中的计数器值进行比较之前处理数据。

下文将详细介绍这些高级步骤的执行情况。

图 2A 和图 2B 示出了图 1A 所示的传输解码器 110 的典型执行过程的高级功能框图。下文参照图 2A 和 2B ，介绍传输解码器 110 中的总体数据流动情况。

如图所示，图 1A 的通道接口 112 与通道数据处理器（ CDP ） 210 相连。通道接口 112 还向 NRSS 接口 212 提供数据，以便用 NRSS 智能卡 230 加密。在此特结合 NRSS 委员会（ 4/95 ）的条件存取 EIA 标准 2.6 版关于 NRSS 智能卡的论述以为参考。

通道数据处理器 210 被提供用于预处理（如检测、同步、确认等）所接收数据流中的包。处理完毕后，通道数据处理器 210 将输出数据发送到传输处理器（ TPROC ） 214 中供进一步处理之用。传输处理器 214 根据下文详细描述的程序标识表进行多种处理，例如语法分析等。

传输处理器 214 与传输解码器 110 中的多个功能块进行交互。这些功能块包括：存储器控制器 216、时钟管理器 218，个人数据处理器 220 和高速数据端口 222。如上所述，微处理器 116 能对多个芯片寄存器直接存取。这种存取是通过微接口 217 和内部微总线 223（如图 2B 所示）获得的。存储器控制器 216 除了与传输处理器 214 连接外，还与外部存储器 114 和视频及音频解码器（224、226 和 228）相连。

现在介绍一下与单个功能块有关的数字数据流的处理细节。

通道数据处理器（CDP）210

值得指出的是来自通道接口 112 中的数据流经过可能的缓冲后（图中未给出）被送往通道数据处理器 210。

通道数据处理器 210 对接收到的数据流执行多种预处理步骤。具体来说，通道数据处理器 210 决定数据流中字节和帧的边界，并将接收到的数据流转换为 8 位的并行格式。在本发明的典型实施例中，在通道数据处理器 210 后在传输解码器 110 中进行的操作使用字节格式进行。

为了确定字节和帧的边界，通道数据处理器 210 与传输包首部包含的帧模式同步。具体来说，这种同步是通过搜索 MPEG-2 的传输流同步字节（sync_byte）（例如：47H）并确认它在可编程数目的连续传输包开始处发生而实现的。

在此过程中，通道数据处理器（210）核对同步字节，将数据流锁定并继续确认包。当同步位的搜索在进行中时，在 CDP 状态寄存器中设置搜索状态位（Search_State）（图中未给出）。一旦发现了同步字节并且对连续发生的确认在进行之中，同步状态位（Sync_State）就设定了。最后，当符合预先确定的搜索标准时，设定锁定状态位（Lock_State）。在本发明的典型实施例中，搜索标准是三个连续的确认过的包。同样，指示带有损坏同步位的包数目的同步滞后值被设定。对通道数据处理器

210 编程使这些同步位在宣布损失锁定状态之前就能通过。当这个滞后值被超过时，锁定的损失被宣布。通道数据处理器 210 然后发出一个锁定损失中断并且从锁定状态过渡回搜索状态。

另外，如果通道接口 112（如图 1A 所示）提供一个 CHPSTRT 信号，传输解码器 110 可能被设置以使用此信号作为一种手段来与数据流同步，方法是将通道数据处理器帧和同步控制寄存器中的帧模式位（Framing_Mode）设为逻辑值“1”。在此情况下，当 CHPSTRT 被确定后，在一个传输包时间内可获得同步。

另一种情况是，如果通道数据处理器帧和同步控制寄存器中的忽略同步位被置为逻辑值“0”，则通道数据处理器 210 在相对于 CHPSTRT 信号过渡过程的预定字节（第一个字节）处校验同步位的完整性，并当校验失败时宣布锁定状态。如果忽略同步（Ignore-Sync）设置为“1”，则通道数据处理器 210 不校验同步的完整性。

取得了锁定状态后，通道数据处理器 210 的输出被传输到传输处理器 214 并由其进行处理，后者以字节的格式对数据进行处理（例如，过滤、语法分析、格式化等）。

传输处理器（TPROC）214

由传输处理器 214 产生的信号控制数据流动去往：存储器控制器 216、微接口 217、时钟管理器 218、个人数据处理器 220、和高速数据端口 220。

在本发明的典型实施例中，传输处理器 214 在程序标识表的控制下处理数据流。如系统部分所述，一个程序标识是传输流首部中一个 13 位的字段，表明存储在包有效数据中的数据类型。一些程序标识值被分配，另一些则被保留。程序标识表的另一些细节在上文提到的 MPEG-2 规范中的系统部分的 3.3 节给出。

根据程序标识表，传输处理器 214 能同时处理多达 32 个用户可选择的程序标识。传输解码器 110 所处理的程序标识在程序标识表中均有规定。程序标识表中的位字段指定了各种数据处理选项。

在操作过程中，程序标识由传输处理器 214 从刚刚到达的传输包的首部中抽取出来，并同时与程序标识表中的所有项目比较。一旦发生匹配，就根据匹配程序标识表项中设定的选项对包中的数据进行处理。如果程序标识表项没有匹配的，就放弃该包。

有效部分格式位没有设定为放弃（000）的程序标识表中每一个地址都与一个通道号相连。存储器控制器 216 与传输处理器 214 相连，负责向 RAM 中特定的缓冲区（通道）传输数据。关于存储器控制器 216 的操作问题在下文中与图 2、3、4 有关的部分具体讨论。

在本发明的典型实施例中，保存于传输处理器 214 中的程序标识表由微处理器 116 在初始化过程中通过微总线 223 利用软件设置。

关于程序标识表项的格式，除了其他信息外，一个程序标识表项还包括：1) 一个 13 位的程序标识，和 2) 一个 1 位的 PCR 程序标识字段。后者一旦设定，就表明对于被解码的程序来说，此程序标识带有 PCR。一个典型的程序标识表项还包括一个 1 位的打包的基本流 HDR 位，它一旦设定，就使得与打包的基本流首部中的“流标识”对应的 RAM 缓冲区地址储存于指针 FIFO 中。当存储器控制器 216 向指针 FIFO 中写入数据时，它发出一个中断。这个中断又被称为一个 DMA_MARK 信号，用于提醒微处理器 116 发送流标识（Stream_ID）字节。在本发明的典型实施例中，以基于表中位置的数字顺序向程序标识表项分配通道号。例如，向第一项分配 DMA 通道 0，向下一项分配 DMA 通道 1 等等。

传输处理器 214 表明带有程序时钟参考程序标识的传输包应该由时钟管理器 218 通过在程序标识表项中放入 PCR_PID 信号进行处理。时钟管理器（218）监视带有 PCR_flag（标志）的包的适应字段，并抽取 PCR。

传输处理器 214 根据包的程序标识将它们分开，然后除去传输包的首部，并结合存储器控制器 216 将程序标识包的有效部分（例如，打包的基本流包的部分）分别存储在外部存储器 114 中指定通道的顺序排列存储区字段。存储器 114 中的数据表示打包的基本流包。

传输处理器 214 还处理从主微处理器 116 向外部存储器进行的写操作。

存储器控制器 216

存储器控制器 216 将传输包中经过语法分析的有效部分存储在外部存储器 114 中，并根据需要向视频处理器 224、多达两个音频处理器 226 和 228 以及主微处理器 116 或微控制器提供视频、音频、PSI 和其他数据（如个人数据）。

图 3 给出了适用于图 2A 和图 2B 中显示的传输解码器 110 的存储器控制器的典型功能框图。如图 3 中所示，存储器控制器 216 通过 FIFO310 从传输处理器 214 中获得数据。存储器控制器 216 通过由微输入 FIFO314 和微输出 FIFO316 组成的微接口 217 从微处理器 116 中取得和传输数据。值得指出的是，图 2A 和图 2B 的存储器控制器 216 外部的微接口 217 在图 3 中也给出了，以便清楚地表示它与 FIFO314 和 316 的相互关系。

存储器控制器 216 还包括一个流程控制单元 318，它包括芯片存储器、一个 DMA 控制器 320、数据路径（Datapath）逻辑 322，以及 用于向视频和音频解码器（224、226 和 228）分别发送数据的。

存储器控制器 216 将来自下列两个来源中的数据写入外部存储器：其一为传输处理器 214（通过 FIFO310 中的传输端口），其二为微处理器 116（通过 FIFO314 中的微端口）。

存储器控制器 216 从外部存储器中读取数据，并将其提供给以下四个部件：1) 视频接口（通过视频 FIFO324），2) 音频接口 A（通过音频接口 A 的 FIFO326），3) 音频接口 B（通过音频接口 B 的 FIFO328），4) 微控制器 116（通过微输出 FIFO316）。

外部存储空间被划分为多个分离的、互不重叠的缓冲区。其中每一个供一个 DMA 通道使用。可使用的通道多达 32 个之多。通道号与写入数据两个 FIFO 源或从四个目标 FIFO 中读出的数据相关连。

由传输处理器 214 提供并写入传入端口 FIFO310 的数据带有一个通道号标记。其它 FIFO 中每一个的通道号由微处理器 116 在 DMA 接口寄存器中定义。微处理器 116 为每一个被使用的通道设置一个 128 位的 DMA 通道配置寄存器（通道寄存器）。此通道寄存器含有由起始和结束地址定义的缓冲区空间等信息，而不管该缓冲区被设为 FIFO（读写指针互相缠绕，有时又叫环形队列）还是队列（ Queue ）（读写指针不会缠绕在一起），等等。

如果通道寄存器允许，DMA 控制器 320 能在当前使用的缓冲区已满时获得一个新的缓冲区。这种特性通常只适用于那些存储为主微处理器 116 所用的数据的缓冲区，如 PSI 区字段。这些新缓冲区的起始和结束地址储存在新的缓冲 FIFO 中。只有微处理器 116 能向此 FIFO 中写入新的缓冲区。已使用过的缓冲区指针和其他类型的指针由 DMA 控制器 320 写入指针 FIFO 中。在本发明的典型实施例中，只有微处理器 116 能读取指针 FIFO。这些指针被微处理器 116 用于管理对缓冲区数据的读取和向新的缓冲 FIFO 返回使用过的缓冲区。图 4 中的指针 FIFO 项流程图给出了

在所有可能的信号条件、通道寄存器设置和缓冲区条件下指针 FIFO 的项。

在接下去的图 3 中，DMA 控制器 320 和数据路径 322 为外部存储器提供所有的接口信号。流程控制单元 318 通过向 DMA 控制器 320 提供适当的控制信号，使其为六个特定的接口 FIFO 中的某一个服务。这些信号包括:

1. DMA 控制器开始信号:

本信号命令 DMA 控制器 320 开始为使用诸如 DMA 控制器操作、DMA 输入 FIFO 选择和 DMA 输出 FIFO 选择之类命令的特定 FIFO 服务。同时，流程控制单元 318 读出 128 位特定通道寄存器的内容，并使其在通道寄存器（内部存储器）的数据总线上可用。 DMA 控制器 320 将此 128 位的字读入它的内部寄存器，然后流程控制单元 318 被释放以根据内部存储器的需要执行其他的读/写操作。

2. DMA 控制器操作信号（ 1 位）:

该位发出一个读或写的信号，由接口 FIFO 上的 DMA 控制器执行。

3. DMA 输入 FIFO 选择信号（ 1 位）:

发出信号选择两个输入 FIFO （ 310 或 314 ）中的某一个提供服务。

4. DMA 输出 FIFO 选择信号（ 2 位）:

发出信号选择四个输出 FIFO （ 316 、 324 、 326 或 328 ）中的某一个提供服务。

5. 在请求/通道数据/通道号/DMA 控制器中的通道数据停止:

一般来说， DMA 控制器 320 持续为 FIFO 服务，直到因如接口 FIFO 满或外部存储器中无数据等原因不能继续服务为止。一旦这种情况发

生， DMA 控制器 320 分配请求的通道数据给流程控制单元 318 ，并且将通道寄存器数据和通道号数据提供给流程控制单元 318 。除非特定的 FIFO 需要急救行动（例如，因为它正接近满或空的状态），流程控制单元 318 使用通道数据更新通道寄存器存储器（例如，被更新的信息包括读写指针等），并指示 DMA 控制器 320 为回路中的下一个 FIFO 服务。

有些时候，流程控制单元 318 可能需要向 DMA 控制器 320 发送一个 DMA 控制器停止命令，作为对向另一个 FIFO 提供服务的需要作出的响应。该 FIFO 可能是拥有更高优先级的输入 FIFO 。这个命令指导 DMA 控制器 320 停止为当前正被其服务的 FIFO 服务，并将通道数据写入流程控制单元 318 。

DMA 控制器 320 使用以下信息更新通道寄存器：读指针、写指针和 DMA 控制器 320 停止为 FIFO 服务处的其他信息。由于流程控制单元 318 将这些信息写入了内部通道寄存器存储器， DMA 控制器 320 可以从中断点开始恢复为被中断的 FIFO 提供服务。

考虑到 DMA 控制器参与本发明不连续状态部分的整体运作过程，当包括一个流标识的字节的下一个 PES 包被 TPROC 214 写入存储器控制器 216 时，流程控制单元 318 引起 DMA_MARK 信号变高。 DMA_MARK 信号的过渡触发微处理器 116 处的中断，并且打包的基本流 包首部的地址被存储在微处理器已存取的输入指针 FIFO 中。由于这个打包的基本流首部可能有一个时间标记，所以对由不连续性指示位产生的标志位作出响应的微处理器 116 在储存的打包的基本流 首部中搜索时间标记。一旦发现，会检索此时间标记值。微处理器 116 基于检索到的时间标记值设置定时器中断。

再参看图 3 ，在本发明的典型实施例中，流程控制器单元 318 的内部存储器包括四个 64 × 32 位的块，每个块都分配给通道寄存器、新缓冲 FIFO 和指针 FIFO 。

在本发明的典型实施例中，每一个通道寄存器有 128 位并且最多有 32 个通道。所以有多达 128 × 32 个位以多个 128 位的格式分配给通道寄存器。

32 位的字被写入新缓冲区和指针 FIFO 或从它们中读出。至少 128 × 32 位的内存（即 128 个字）可用于新缓冲区和指针 FIFO 中。另外，如果使用的通道少于 32 个，通道寄存器中未使用的空间被分配给新的缓冲区和指针 FIFO 。

以下是 32 个 DMA 通道和 16 个 DMA 通道的存储器分配示例。

1) 32 个被使用的通道的存储器分配

新缓冲区 FIFO = 64 × 32 位

指针 FIFO = 64 × 32 位

通道寄存器 = 128 × 32 位

2)16 个被使用的通道的存储器分配

新缓冲区 FIFO = 64 × 32 位

指针 FIFO = 128 × 32 位

通道寄存器 = 128 × 16 位

时钟管理器 218

图 2A 和 2B 中所示的传输解码器 110 中的时钟管理器 218 使用连续 PCR 值同步本地 27MHz 的信号（由时钟信号电路 122 产生）。时钟管理器 218 中的计数器值被称为系统时间常数（ STC ）值。在本发明的典型实施例中，微处理器 116 通过内部微总线 223 存取计数器中的 STC 值。

时钟管理器 218 中的计数器定期与传输数据流中的 PCR 进行同步。当遇到数据流中的第一个 PCR 时，该 PCR 被“挤载”入计数器并成为初始 STC 值。当后来的 PCR 值被接收时，将它们与 STC 值相比较，以校

正时钟信号电路 122（例如， VCXO ）的频率。在本发明中，此过程是通过使用在下文中会详细介绍的数字模拟转换器（ DAC ）实现的。一旦进行了相位错误校正，通常是将新的 PCR 值作为 STC 值载入。

一般每帧产生一次 PCR 值。在 MPEG-2 的标准中，至少每三帧内（大概十分之一秒）会产生一次 PCR 值。

典型的情况是，外部视频和音频解码器（ 118 和 120 ）通过更新外部解码器中的 STC 值与传输解码器 110 保持同步，该过程的实现基于时钟管理器 218 中保存的 STC 值。视频和音频解码器（ 118 和 120 ）通过将数据和接收到的打包的基本流 数据流中的时间标记相比较，使用储存值来确定何时解码或显示数据。值得指出的是，在本发明的典型实施例中，传输解码器 110 、视频解码器 118 和音频解码器 120 都接收时钟管理器（ 218 ）产生的同一个 27MHz 的时钟信号。

传输解码器 218 包含对产生本地时钟信号的硬件支持。该信号对于某一个选定的程序来说是锁定在系统时钟信号上的。如上文所述，在特定的 PCR_PID 上接收的 PCR 值在串行接口上可为其他设备所用。

时钟管理器（ 218 ）包括一个系统时间计数器（ STC ）（图中未给出）、一个系统时间时钟寄存器（图中未给出）、一个程序时钟参考（ PCR — Program Clock Reference ）寄存器（图中未给出）、一个当前 STC 寄存器（图中未给出）、一个报警时钟中断寄存器（图中未给出）以及一个锁存从与选定的程序标识（ PID ）有关的首部中抽取的 PCR 值的逻辑部件（图中未给出）。它还包括串行 PCR 接口和用于控制外部系统时钟信号控制回路的 DAC 寄存器。

在本发明的典型实施例中， STC 是由 9 位基部和 33 位扩展部组成的 42 位计数器。

STC 基部将名义上的 27MHz 系统时钟输入信号分为 300 份。结果得到的 90KHz 信号驱动 STC 扩展。在任何时候， STC 都能被在微处理器 116 上运行的软件写或读。当遇到一个不连续性指示位后，可以将接收到的 PCR 值自动加载到 STC 中。

如上文所述，可以通过在程序标识表项中设定 PCR_PID 位，从刚到达的位流中自动抽取 PCR。处理接收到的 PCR 取决于软件设定的选项和接收 PCR 时当前传输包的不连续状态是否存在。

当包含 PCR_PID 的传输包被接收时，和传输首部中的不连续指示位为逻辑值 1 ，可以确认是不连续状态。该不连续状态将一直持续下去，直到包含 PCR_PID 的传输包被接收时不连续性指示位为逻辑值 0 ，或者当在 PCR_PID 包中发现下一个 PCR 为止。

如果设置为有效，当 PCR 被接收且不连续状态存在时，时钟管理器 218 自动将接收到的 PCR 载入 STC 中。此时，一个中断被发送到主微处理器 116 。

时钟管理器 218 包括一个初始化位。该位被软件置为 1 时，则无论不连续状态存在与否，下一个被接收到的 PCR 都将被载入 STC 中。然后初始化位被时钟管理器 218 自动清除。只有当 STC 不包含有效的 PCR 时（即，当选择了新的程序标识时），才设置初始化位。

如果没有不连续状态存在或自动 STC 载入功能被置为无效，接收到的 PCR 被储存在 PCR 寄存器中，并且可选择向主微处理器 116 发出一个中断。在获得 PCR 的同时， STC 的当前状态被保存在 STC 寄存器中。储存下来的 PCR 和 STC 值被用于计算错误信号，从而控制外部系统的时钟信号（ XOCLK ）。

当接收到的传输包包括一个与 PCR_PID 等同的 PID（程序标识）和一个 PCR 时，此 PCR 被锁存在 PCR 寄存器中。此时 STC 计数器的当前

内容也被锁存在 STC 寄存器中，并且向主微处理器 116 发出一个中断。将两个值都锁存起来就消除了由于中断服务例程中存在变化的等待时间而产生的不确定性。在本发明的典型实施例中，两个寄存器都包括一个 9 位的基部和一个 33 位的扩展部。基部在 300 模块中，扩展部是用二进制表示的。在寄存器的内容被用于任何计算之前，它们可以被转化成二进制数。转化方法是将扩展部乘以 300，然后将与基部相加。

当前的 STC 寄存器是一个只读寄存器，它允许微处理器 116 在任何时间获得 STC 的当前内容。通过外部微总线 124，微处理器 116 能用这个值初始化视频解码器或音频解码器的 STC 。

如上文所述，传输解码器 110 包括一个 10 位的数字模拟转换器（ DAC ） 230 ，它在时钟管理器 218 的时钟校正控制回路中使用。DAC230 是只读的，也就是说，软件在存储器中存有最后载入 DAC 的值的副本。在运作中， DAC230 被用于为外部谐振晶体电压控制振荡器 VCXO122 产生控制电压。后者是系统时钟频率（名义上是 27MHz ）的来源。通过比较锁存的 PCR 和 STC 值，软件能确定本地时钟信号是否需要校正。例如，软件能获取 PCR 和 STC 之间的差别，并将这个错误条件加到 DAC230 的输入寄存器的当前内容中去。

总体来说，本发明的不连续部分包括以下内容:

1) 通道数据处理器（ CDP ） 210 核查接收到的传输包中的同步字节，以使该包生效。传输处理器（ TPROC ） 214 对传输首部进行语法分析，并将包的有效内容传送到存储器中的适当位置。根据传输首部中的程序标识值，数据被发送到存储器中的不同 FIFO 缓冲区（通道）中。

2) 因为接收到的不连续性指示位不必和下一个 PCR 值在同一个包中，所以接收到下一个 PCR 值时， TPROC （ 214 ）将“挤载”时钟管理器（ 218 ）中的计数器。

3) 当包括一个流标识字节的下一个打包的基本流包被传输处理器 214 写入存储器控制器 216 中时， DMA_MARK 信号变高。这就导致存储器控制器 216 向微处理器 116 发出中断并将打包的基本流 包首部的地址储存在一个输入指针 FIFO 中。这个打包的基本流 首部可能有一个时间标记（可能是解码时间标记或显示时间标记）。对来自存储器控制器 216 的中断作出响应的微处理器 116 在打包的基本流 首部中搜索时间标记，一旦发现，就检索时间标记值。

4) 然后，微处理器 116 在时间标记值前略短于一帧的时间处设置一个定时器中断。

5) 当此定时器中断发生时，微处理器 116 通过外部微总线 124 向视频和音频解码器发送新的 PCR 值，从而给了解码器足够的时间处理数据。这种处理发生在时间标记与视频或音频解码器中的计数器值比较之前，并且不会干扰与前一个 PCR 值相参照的数据的处理。

视频/音频解码器 224 、 226 、 228

如上文所述，一旦传输包被接收、处理和存储起来，存储器控制器 216 从存储器 114 中读出打包的基本流包，并将它们发送到外部视频和音频解码器中。然而，在打包的基本流 包到达外部解码器（ 118 和 120 ）前，它们被内部视频和音频处理器（ 224 、 226 和 228 ）处理。

这些内部处理器的一个重要功能在于确保离开传输解码器的数据流与外部解码器兼容。也就是说，不同的外部解码器有不同的数据需求。

例如，一些现在市场上可见的外部视频解码器能接受包括首部和有效部分的打包的基本流包流。这些流的形式主要与它被读出存储器时相同。另一方面，有些外部解码器只能接受对基本流（例如，打包的基本流 包的有效部分而非首部部分）的处理要求。另外，后一种类型的解码器除了要求有数据流外，还要求相关的控制信息，如时间标记和字节计

数等。由 C - Cube 制造的 CL9100 就是这样一种视频解码器。在此特参考其规格说明——CL9100 多模式视频解码器用户手册（94 年 10 月）——加以介绍。由此，视频处理器 224 可通过微处理器 116 进行设置，以格式化输出的数据流，使之与选定的外部解码器的要求兼容。

在处理 C - Cube CL9100 等外部视频解码器的接口方面，视频处理器 224 对打包的基本流 包的流进行处理，从而去除打包的基本流 的首部并将时间标记信息从打包的基本流 首部中取出来。另外，视频处理器 224 通过使用计数器（图中未给出），对发送给外部解码器的打包的基本流包保持字节计数。那么，视频处理器 224 包含时间标记信息，保持字节计数并将打包的基本流 包的有效部分传输给外部视频解码器。为了获得传送给外部解码器的时间标记和字节计数信息，视频处理器 224 向微处理器 116 发出一个中断。后者进行响应时，会通过内部微总线 223 从视频处理器 224 中读取时间标记和字节计数信息，并通过外部微总线 124 将这个控制信息发送到外部视频解码器中。

然而，有时在微处理器 116 能对来自视频处理器 224 的中断作出响应之前，另一个带有时间标记信息的打包的基本流 首部就到达了。例如，当微处理器 116 在很长的时间里被更高优先级的任务占用时，这种情况就可能发生。在这种情况下，视频处理器 224 将时间标记信息和状态一起储存起来。具体来说，视频处理器 224 会保存一个状态寄存器，表明微处理器是否“丢失”了时间标记。在本发明的典型实施例中，微处理器 116 最后才开始读取时间标记信息，如果它象状态中表示的那样丢失了一个或多个时间标记，那么它将只能恢复最新的时间标记。

另外，既然时间标记信息和字节计数信息将被提供给外部解码器，那么当微处理器 116 丢失了一个时间标记时，视频处理器 224 保持一种累积式的字节计数，因为许多数据标记可能在微处理器 116 读取信息之前就到达了。然后，微处理器 116 读取最新的时间标记和累积式的字节

计数,并将这些信息发送到外部视频解码器中。接下来，状态寄存器和字节计数器可以被重置。

如图 5A 中的流程图所示，接受到的数据的首部被分离并进行语法分析（步骤 510 ）。时间标记信息被存储并且字节计数被保存（步骤 510 ），由视频处理器 224 向微处理器 116 发出一个中断，提醒它时间标记信息可供读取（步骤 514 ）。如果附加的时间标记信息在微处理器读取前一个时间标记信息之前就已收到，新时间标记信息被存储，字节计数器继续积累（步骤 516 ），视频处理器 224 向微处理器 116 发出另一个中断，否则，微处理器读取状态寄存器、时间标记信息和字节计数信息并将它发送到外部视频解码器中（步骤 518 ）。

调节某个外部解码器的另一个例子是使用位速率限制音频解码器，例如 Zoran ZR38500 AC3 音频解码器。在此特引用其技术说明书（ 1994 年 10 月）以供参考。一般说来，大多数音频解码器提供一个数据请求，表明它们能接受音频数据，然后传输解码器 110 通过音频处理器（ 226/228 ）提供音频数据，直到外部音频处理器取消此请求。换句话说，这里存在着一种信号交换机制。

然而，位速率限制解码器，例如 Zoran AC3 音频解码器，要求音频 AC3 数据能作为音频数据帧发送，每一帧在首部中都包括一个帧时间和位速率。以高于预先确定的位速率发送数据可能会引起内部缓冲区溢出，从而导致数据丢失。进一步说，这些位速率限制解码器并不提供数据就绪的信号以表明它们已经准备好接收附加的音频数据。

所以，和视频解码器 224 一样，音频处理器 226 和 228 可被设置以调节选定的外部音频解码器。值得指出的是：在本发明的典型实施例中，除非是对于微处理器 116 可以存取的地址，音频处理器（ 226/228 ）在本质上是相同的。

因此，当音频处理器（226/228）被编程用于与该类音频解码器接口时，它们假定解码器一直在发送数据请求信号。然而，当打包的基本流包音频数据从存储器控制器216中提供给音频处理器（226/228）时，接收到数据的打包的基本流 首部被音频处理器进行语法分析。音频数据的帧大小和位速率被包含在经分析过的首部中。根据帧的大小和位速率，音频处理器（226/228）能确定帧边界（即帧时间）。有了帧时间信息，音频处理器（226/228）设置帧定时器，使得音频数据可用时，定时器被设定，一帧数据被触发到外部音频解码器及音频处理器（226/228）中，然后暂停，直到帧定时器超过时效。当定时器超过时效时，循环继续下去，每次一个音频帧，而另外的帧可用于向外部音频解码器中输送附加帧。控制信息，例如时间标记，由微处理器恢复并通过外部微总线发送出去。

如图 5B 中的流程图所示，从存储器中恢复打包的基本流包（步骤542），去除接收到数据的首部并进行语法分析（步骤530），确定一个帧时间(步骤532),定时器开始工作(步骤534),音频处理器（226/228）触发出一帧数据（步骤536），音频处理器（226/228）暂停，直到定时器超过时效（步骤538），尽管如图中所示，步骤542、530和532能在暂停期间执行。然后，此过程延续下去。

尽管在上述特定的实施例中，对跟踪 MPEG-2 传输数据流不连续性的系统时间常数进行更新的方法和设备进行了描述。但本发明并不局限于所示的细节。另外，无需脱离权利要求的范畴和本发明的精神实质，可实现细节的多种变化。

说 明 书 附 图

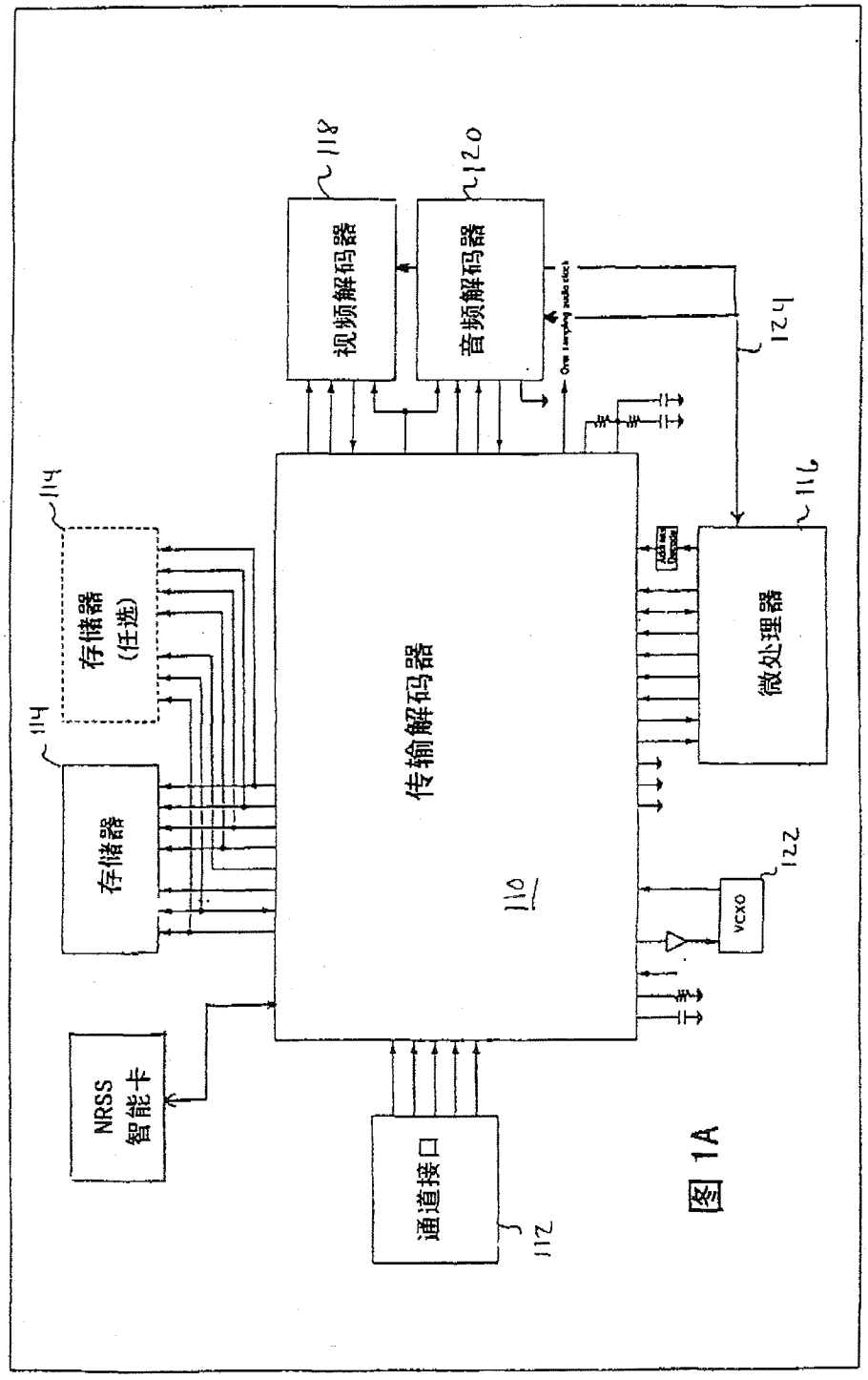

视频解码器
118
音频解码器
120
121
传输解码器
110
微处理器
116
存储器
(任选)
114
存储器
114
VCXO
122
NRSS
智能卡
通道接口
112

图 1A

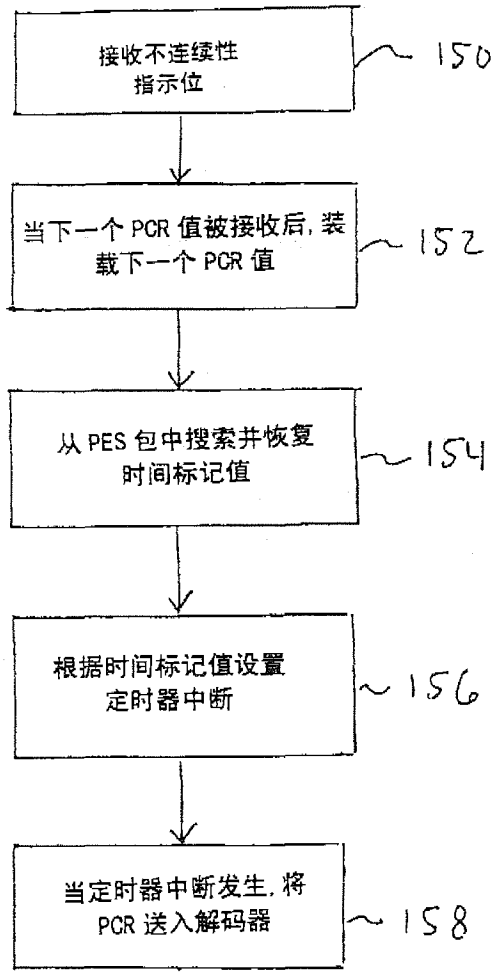

接收不连续性
指示位
150
当下一个 PCR 值被接收后，装
载下一个 PCR 值
152
从 PES 包中搜索并恢复
时间标记值
154
根据时间标记值设置
定时器中断
156
当定时器中断发生，将
PCR 送入解码器
158

图 1B

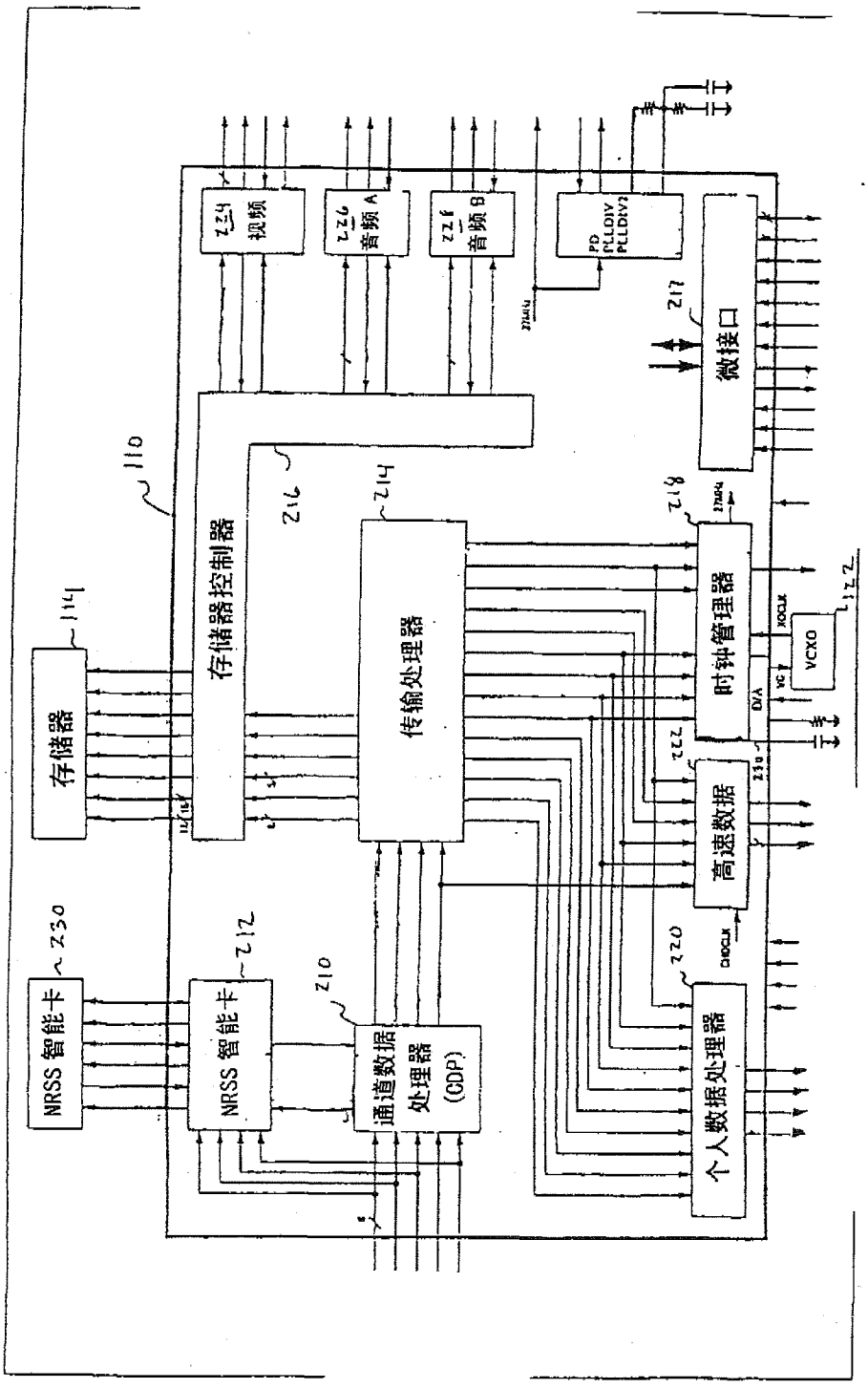

110
存储器
存储器控制器
传输处理器
时钟管理器
微接口
高速数据
个人数据处理器
通道数据处理器 (CDP)
NRSS 智能卡
NRSS 智能卡
视频
音频 A
音频 B
VCXO

图 2A

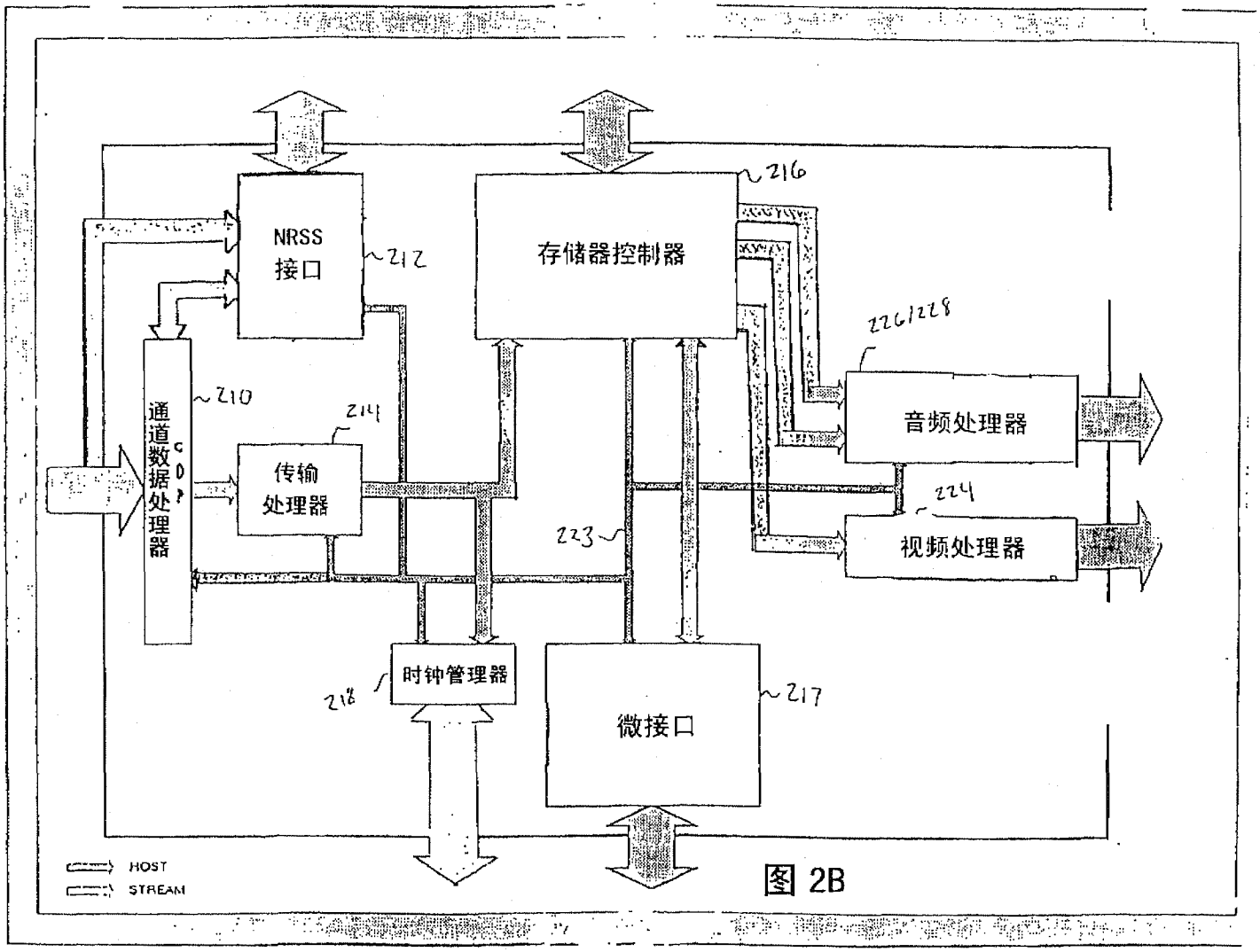

NRSS
接口
212
存储器控制器
216
226/228
通道数据处理器
CDP
210
传输处理器
214
音频处理器
视频处理器
224
223
时钟管理器
218
微接口
217
HOST
STREAM

图 2B

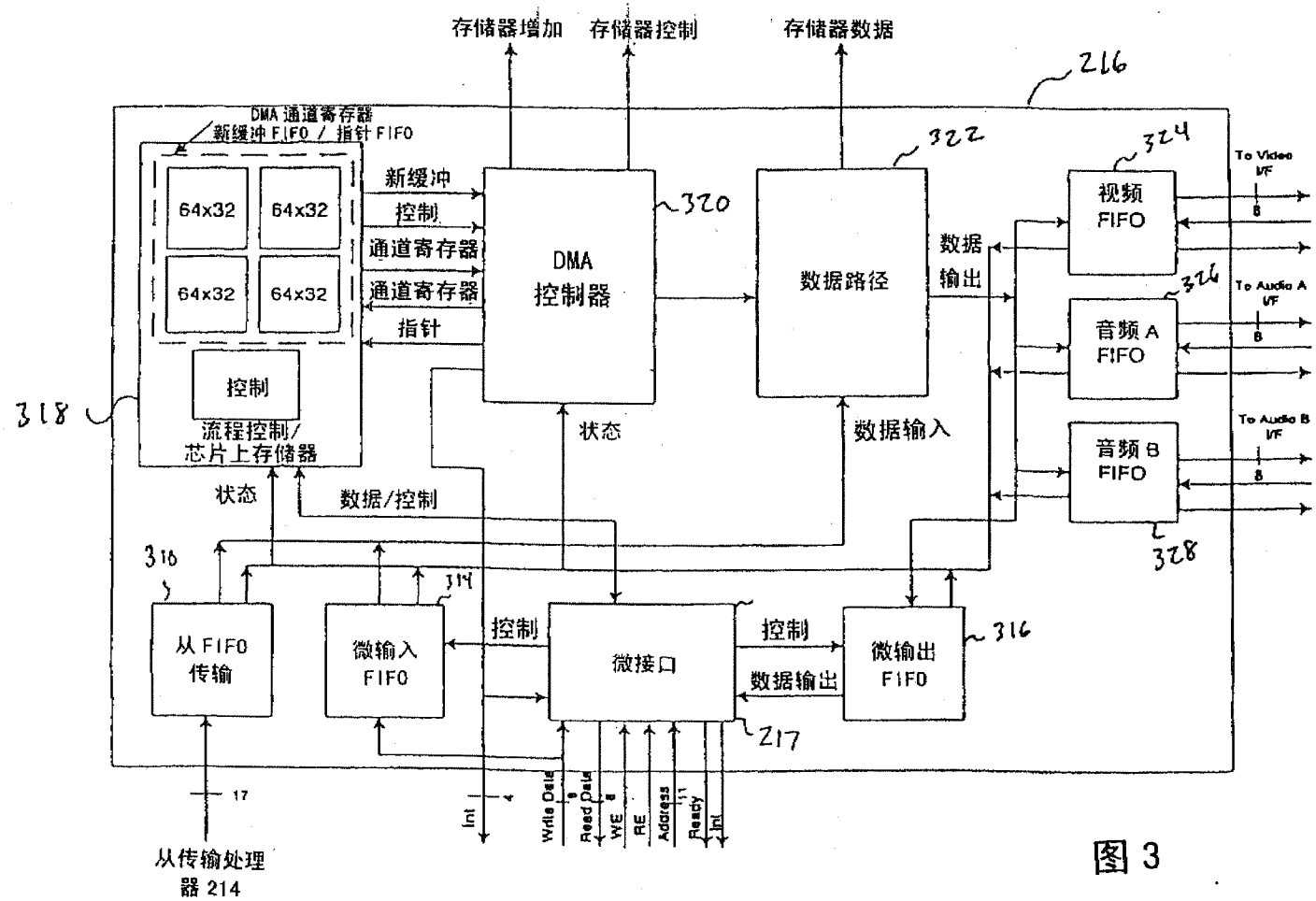
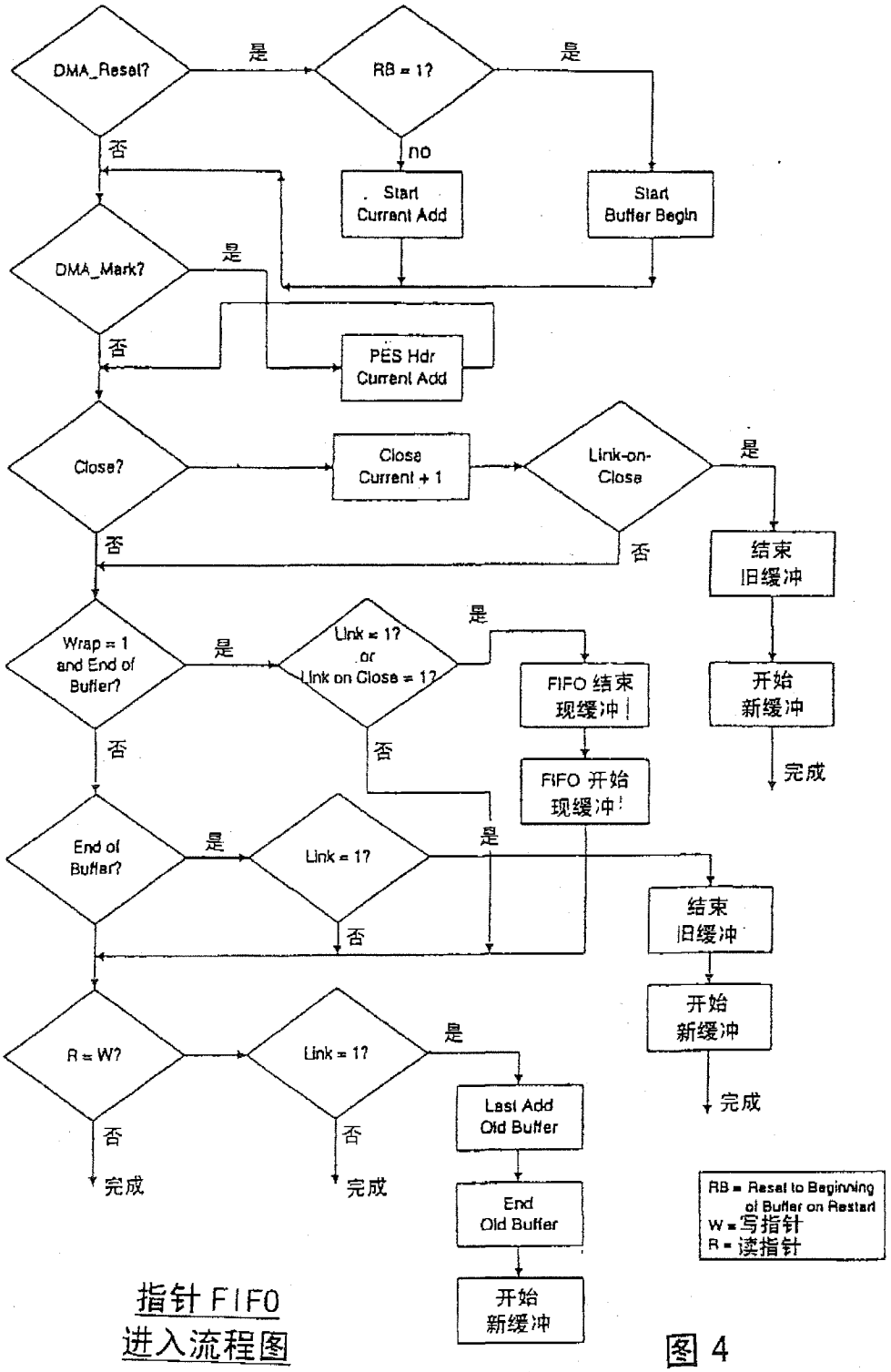

存储器增加
存储器控制
存储器数据
216
DMA 通道寄存器
新缓冲 FIFO / 指针 FIFO
64x32
64x32
64x32
64x32
控制
流程控制/
芯片上存储器
318
新缓冲
控制
通道寄存器
通道寄存器
指针
DMA
控制器
320
322
数据路径
数据
输出
数据输入
状态
数据/控制
状态
视频
FIFO
324
音频 A
FIFO
326
音频 B
FIFO
328
To Video
I/F
To Audio A
I/F
To Audio B
I/F
310
从 FIFO
传输
314
微输入
FIFO
控制
微接口
217
控制
数据输出
微输出
FIFO
316
17
从传输处理
器 214
Int
Write Data
Read Data
WE
RE
Address
Ready
Int

图 3

存储器控制器锁定



指针 FIFO
进入流程图

图 4

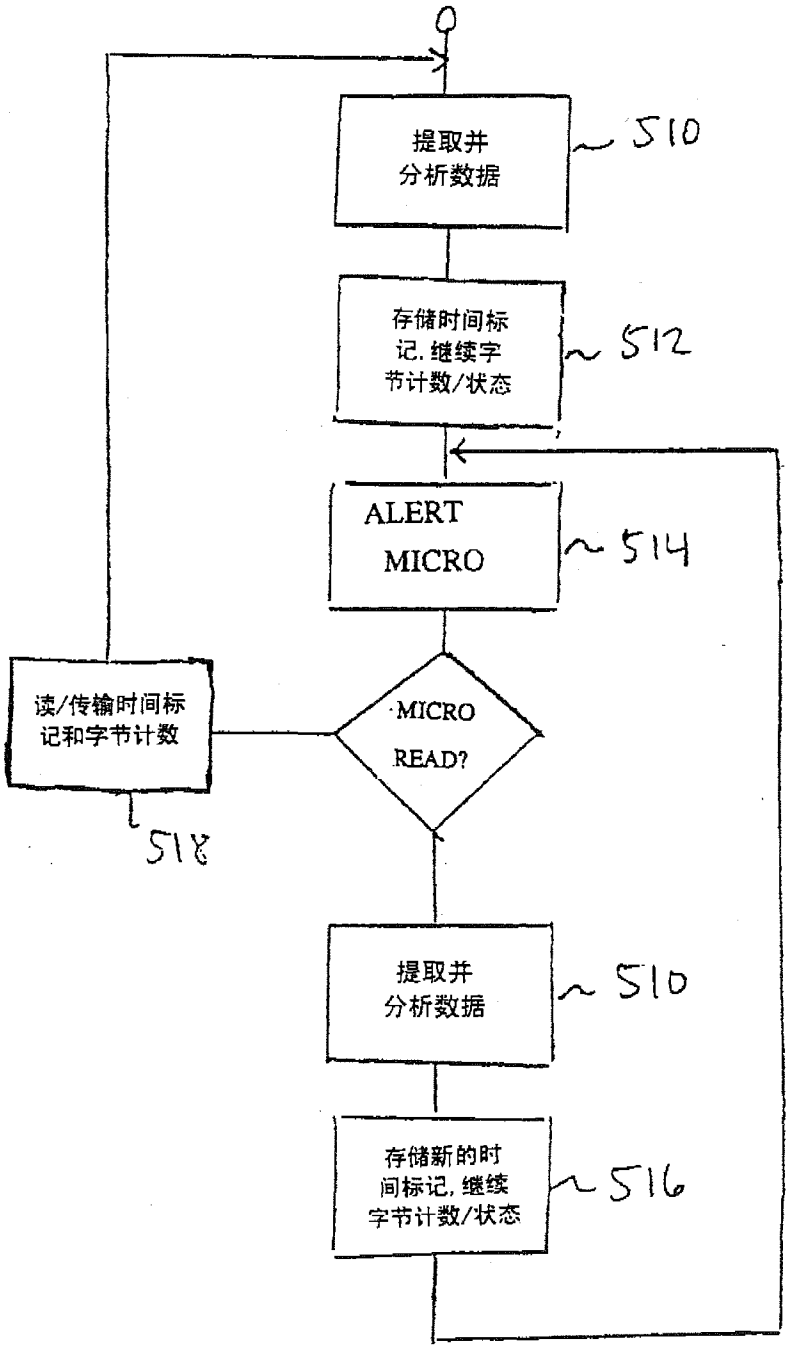

提取并
分析数据
510
存储时间标
记. 继续字
节计数/状态
512
ALERT
MICRO
514
MICRO
READ?
读/传输时间标
记和字节计数
518
提取并
分析数据
510
存储新的时
间标记, 继续
字节计数/状态
516

图 5A

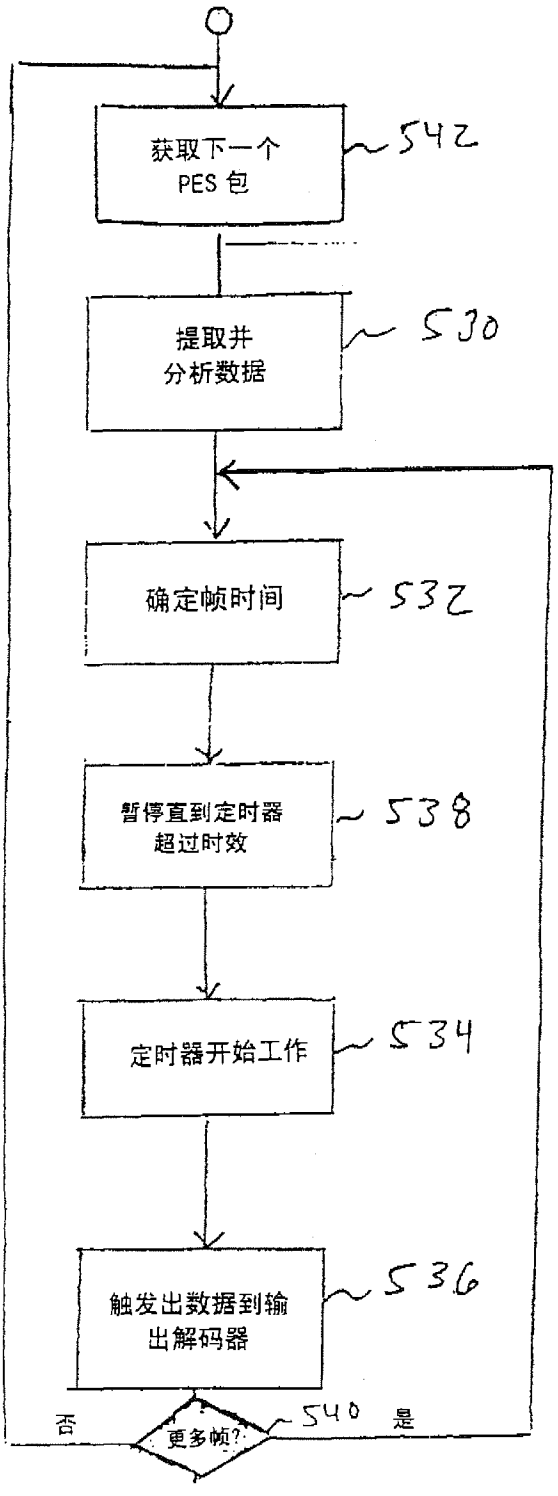

获取下一个
PES 包
542
提取并
分析数据
530
确定帧时间
532
暂停直到定时器
超过时效
538
定时器开始工作
534
触发出数据到输
出解码器
536
更多帧?
540
否
是

图 5B

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	6,266,158	7/24/2001	Hata et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	10-271511	10/09/1998	JP				X

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

拒絶理由通知書

原/

特許出願の番号	特願２００３－５６３２２９
起案日	平成２１年　３月２６日
特許庁審査官	坂本　聡生　　２９５４　５Ｃ００
特許出願人代理人	青木　篤（外　４名）　様
適用条文	第２９条第２項、第３６条、第３７条

この出願は、次の理由によって拒絶をすべきものです。これについて意見がありましたら、この通知書の発送の日から３か月以内に意見書を提出してください。

理　由

ａ．この出願は、下記の点で特許法第３７条に規定する要件を満たしていない。

記

請求項１－１２，１４－２４に記載される発明が解決しようとする課題は、本願明細書（特に第【００１２】段落－第【００１３】段落参照）に記載の「最初のＰフレームが、通常、再構成のために最も重要なものである。しかしながら、既知の符号化方法は、このような重要度の差異を信号伝送する簡単な方法を提供できていない」という課題、又は、「逸脱した画像番号を、喪失した画像フレームの信号として解釈して、喪失と疑われる画像フレームを再構成したりあるいはその再送信を要求したりするために不要な動作を開始してしまう」問題にあると認められるが、請求項１３に記載される発明が解決しようとする課題は、画像の再構成に関する課題を解決するものではなく、両者が同一であるとは認められない。

よって、請求項１－１２，１４－２４に記載される発明、請求項１３に記載される発明は、それぞれの解決しようとする課題が同一でなく、特許法第３７条第１号に規定する関係を有するとは認められない。

また、請求項１－１２，１４－２４に記載される発明の主要部と請求項１３に記載される発明の主要部が同一であるとも認められないから、特許法第３７条第２号に規定する関係を有すると認められない。

さらに、各発明は、特許法第３７条第３号、第４号、第５号に規定する関係のいずれを満たすものとも認められない。

この出願は特許法第３７条の規定に違反しているので、請求項１－１２，１４－２４以外の請求項に係る発明については特許法第３７条以外の要件についての審査を行っていない。

ｂ．この出願の下記の請求項に係る発明は、その出願前に日本国内又は外国において、頒布された下記の刊行物に記載された発明又は電気通信回線を通じて公衆に利用可能となった発明に基いて、その出願前にその発明の属する技術の分野における通常の知識を有する者が容易に発明をすることができたものであるから、特許法第２９条第２項の規定により特許を受けることができない。

記　　（引用文献等については引用文献等一覧参照）

請求項：１－１２，１４－１５，１８－２４
引用文献：１

備考：

引用文献１には、参照画像をＩピクチャとして符号化する手順と、予測画像をＰピクチャとして符号化する手順と、を有し、いずれの画像が予測画像のグループに属するかを示す参照関係情報を作成する点（特に第【００４８】段落－第【００５４】段落参照）が記載されている。

そして、階層符号化技術は周知であり、引用文献１記載の発明の符号化を階層符号化により行うことに格別の困難性は認められない。

ｃ．この出願は、特許請求の範囲の記載が下記の点で、特許法第３６条第６項第１号に規定する要件を満たしていない。

記

（１）請求項１には、「第１のサブシーケンス」、「第２のサブシーケンス」と記載されているが、単に「第１」と「第２」の記載のみでは、それぞれがベースレイヤとエンハンスメントレイヤに対応するのか否かなど、「第１のサブシーケンス」と「第２のサブシーケンス」とが、いずれのサブシーケンスを指すのか不

明確であるため、発明の詳細な説明との対応関係が不明確である。
　他の請求項における「第１のサブシーケンス」、「第２のサブシーケンス」の記載も同様に不明確である。

（２）請求項１には、「どのビデオフレームが前記第２のサブシーケンスに属するのかを指示する情報」と記載されているが、「どのビデオフレームが・・・属する」の記載のみでは、例えば、シーケンス全体のうち第２のサブシーケンスに属するフレームを列挙したような情報であるのか、各ビデオフレーム毎に自らがいずれのサブシーケンスに属するのかを示す情報であるのか、不明確である。

　また、「属するのか」の記載では、第２のサブシーケンスのビデオフレームであることを指示する情報であるのか、第２のサブシーケンスに依存するビデオフレームであることを指示する情報であるのか、不明確である。

　その結果、発明の詳細な説明との対応関係が不明確である。

　請求項１４，２２の記載も同様に不明確である。

（３）請求項４には、「依存性を、少なくともスケーラビリティレイヤの識別子及びサブシーケンスの識別子に基づいて定める」と記載されているが、「依存性を・・・（中略）・・・基づいて定める」の記載では、例えば、依存先のサブシーケンスを、スケーラビリティレイヤの識別子及びサブシーケンスの識別子を用いて示すことを意味するようにも解され、発明の詳細な説明との対応関係が不明確である。

ｄ．この出願は、特許請求の範囲の記載が下記の点で、特許法第３６条第６項第２号に規定する要件を満たしていない。

記

（１）請求項１には、「サブシーケンス」と記載されているが、該記載のみでは、いかなる単位のシーケンスであるのか、不明確である。
　他の請求項における「サブシーケンス」の記載も同様に不明確である。

（２）請求項３には、「ビデオシーケンスを、複数のスケーラビリティレイヤ内に符号化する」と記載されているが、「複数の・・・（中略）・・・内に」の記載では、複数のスケーラビリティレイヤからなる符号化データとして符号化することを意味するのか否か、不明確である。

（３）請求項３には、「第２のサブシーケンスのビデオフレームの前記依存性を、前記第２のサブシーケンスの少なくとも１つのビデオフレームが、高位のスケーラビリティレイヤのビデオフレームと、同じスケーラビリティレイヤ内の他の１つのサブシーケンスのビデオフレームと、を含むグループから予測されるように定める」と記載されているが、「依存性を・・・（中略）・・・予測されるように定める」の記載では、依存して予測することを意味するのか、依存性を示す情報をビデオシーケンス内に形成することを意味するのか、不明確である。

（４）請求項３には、「１つのビデオフレームが、高位のスケーラビリティレイヤのビデオフレームと、同じスケーラビリティレイヤ内の他の１つのサブシーケンスのビデオフレームと、を含むグループから予測される」と記載されているが、「高位」又は「同じ」の記載のみでは、いずれのビデオフレームより「高位」、又は、いずれのビデオフレームと「同じ」であるのか、不明確である。

（５）請求項３には、「高位のスケーラビリティレイヤのビデオフレームと、同じスケーラビリティレイヤ内の他の１つのサブシーケンスのビデオフレームと、を含むグループ」と記載されているが、該記載のみでは、グループが、請求項３に記載（高位の・・・（中略）・・・サブシーケンスのビデオフレーム）の上記ビデオフレームを含むビデオシーケンス全体であるのか、２枚の上記ビデオフレームのみからなるグループであるのかなど、グループがいかなる単位のグループであるのか、不明確である。

（６）請求項３には、「グループから予測される」と記載されているが、「グループ」の記載では、いずれのフレームから予測が行われるのか、不明確である。

（７）請求項１１には、「サブシーケンスのフレームの少なくともいくつかは、時間的に少なくとも部分的に重複」と記載されているが、該記載のみでは、異なるレイヤのサブシーケンスのフレームが重複するのか、同じレイヤのサブシーケンスのフレームが重複するのかなど、いずれのレイヤのフレームといずれのレイヤのフレームが重複するのか、不明確である。

請求項２０の「部分的に時間的に重複」の記載も、同様に、不明確である。

（８）請求項１６には、「喪失した画像番号に対応するいくつもの画像フレームを含む」と記載されているが、「いくつもの」の記載では、「いくつ」であるのか不明確であり、例えば、喪失した画像番号に対応する画像フレームであるのか否か、不明確である。

請求項１７の「いくつもの」の記載も同様に不明確である。

（９）請求項１９には、「独立して復号可能なサブシーケンスを定める」と記載されているが、「定める」の記載では、例えば、サブシーケンスを復号可能なものと見なすなどして、定めることを意味するのか、復号可能なサブシーケンスを探すなどして、特定することを意味するのか、不明確である。

（１０）請求項１９には、「画像番号値を０に設定」と記載されているが、該記載のみでは、符号化の時に設定するのか、復号の時に設定するのか、不明確である。

（１１）請求項２０には、「復号順序における最後の画像フレーム」と記載されているが、「最後」の記載のみでは、シーケンス全体の最後であるのかなど、いかなる単位の最後であるのか、不明確である。

（１２）請求項２０には、「部分的に時間的に重複した画像フレームのグループ」と記載されているが、「時間的に重複した画像フレームのグループ」の記載のみでは、該グループが、時間的に重複した画像フレームを含むシーケンス全体を意味するのか、レイヤが異なるサブシーケンスであって時間的に重複するものを集めたグループであるのかなど、具体的に、いかなるグループであるのか、不明確である。

（１３）請求項２２には、「符号化するコンピュータプログラムであって、・・・（中略）・・・予測されるコンピュータプログラムにおいて、該コンピュータプログラムは、・・・（中略）・・・するためのプログラムコードであって、・・・（中略）・・・されているプログラムコードと、・・・（中略）・・・たものであるプログラムコードと、・・・（中略）・・・ためのプログラムコードと、を含むことを特徴とするコンピュータプログラム。」と記載されているが、該記載のみでは、符号化を実行する主体が不明確であり、例えば、コンピュータに符号化処理を実行させるためのプログラムであるのか否か、不明確である。

また、「プログラムコードと、・・・（中略）・・・プログラムコードと」の記載では、各符号化手順を実行させるためのプログラムであるのか否か、不明確である。

したがって、上記記載では、コンピュータに、請求項２２に記載の各手順を実行させるためのプログラムであるのか否か、不明確である。

請求項２３の記載も同様に不明確である。

（１４）請求項２４には、「ビデオ信号」と記載されているが、該記載では、「

物」の発明であるのか、「方法の発明」であるのか特定できず、不明確である。

（１５）上記理由ｃ．で指摘の点については、同理由に記載の理由により、各請求項に係る発明が不明確である。

なお、補正の際には、新規事項の追加とならないように留意の上、意見書で、各補正が適法である理由について、根拠となる出願当初明細書の記載箇所とともに、主張されたい。

拒絶の理由が新たに発見された場合には拒絶の理由が通知される。

引　用　文　献　等　一　覧

１．特開平１０－２７１５１１号公報

－－

先行技術文献調査結果の記録

・調査した分野　　ＩＰＣ

Ｈ０４Ｎ　　７／２４　　－　　７／６８

この先行技術文献調査結果の記録は、拒絶理由を構成するものではない。

B046260 Dispatch Number: 199591 Mailing Date: March 31, 2009

NOTICE OF REASONS FOR REJECTION

Patent Application No. 2003-563229
Drafting Date: March 26, 2009
Patent Office Examiner: Satoo Sakamoto 2954 5C00
Attorney for Patent Applicant: Atsushi Aoki (and four others)
Applicable Provisions: Article 29, paragraph 2; Article 36; Article 37

It is deemed that the present application should be rejected for the following reasons. Any argument, if any, should be submitted within 3 months from the mailing date of this notice.

REASONS

(a) It is deemed that this application does not satisfy the requirements prescribed in Article 37 of the Patent Law on the following points.

NOTE

It is deemed that the problems to be solved by the inventions claimed in claims 1 to 12 and 14 to 24 are as described in "the first P-frames are usually the most significant ones for the reconstruction. However, known coding methods fail to provide a simple method for signaling such differences in significance" and "the deviating image

numbering is interrupted as a signal of lost image frames and unnecessary actions are started to reconstruct the image frames suspected as lost or to request a re-transmission thereof." However, the problem to be solved by the invention claimed in claim 13 is not to solve the problem regarding an image reconstruction and is not deemed to be the same as those of claims 1 to 12 and 14 to 24.

Therefore, it is not deemed that the inventions claimed in claims 1 to 12 and 14 to 24 and the invention claimed in claim 13 have the same "problems to be solved by the invention" and satisfy the relationship prescribed under Article 37, item 1, of the Patent Law.

Moreover, it is not deemed that the inventions claimed in claims 1 to 12 and 14 to 24 and the invention claimed in claim 13 have the same substantial part and satisfy the relationship prescribed under Article 37, item 2, of the Patent Law.

Furthermore, it is deemed that each invention of this application does not satisfy the relationship prescribed in any of Article 37, Items 3, 4 and 5.

As this application violates the provision of Article 37, an examination was not carried out regarding the requirements other than Article 37 of the Patent Law for the inventions claimed in claims other than claims 1 to 12 and 14 to 24.

(b) The inventions described in the following claims of this

application are deemed ones which could easily have been made, prior to the filing of the patent application, by a person with ordinary skill in the art to which the inventions pertain, on the basis of the inventions described in the following publications distributed or the inventions made available to the public through electronic telecommunications lines in Japan or a foreign country prior to the filing of the patent application and, therefore, are unpatentable under the provisions of Article 29, paragraph 2, of the Patent Law.

NOTE (For the numbers of the cited references, see the List of Cited References)

Claims Nos. 1 to 12, 14, 15, and 18 to 24
Reference No. 1

Remarks:

Reference 1 describes the feature that a step of encoding a reference image as the I-PICTURE and a step of encoding a predictive image as the P-PICTURE are provided to create the reference relationship information showing the images belonging to the group of predictive images (specially refer to paragraphs [0048] to [0054]).

A hierarchical coding technology is well known and it is not deemed that it is particularly difficult to perform encoding in the invention described in Reference 1 using a hierarchical coding.

(c) The description of the claims of this application does

not satisfy the requirements prescribed in Article 36, paragraph 6, item 1 of the Patent Law on the following points.

NOTE

(1) Claim 1 describes "a first sub-sequence" and "a second sub-sequence", however, the sub-sequences indicated by "a first sub-sequence" and "a second sub-sequence" are unclear only with "first" and "second", for example, whether or not the "first sequence" and the "second sequence" correspond to "the base layer" and "the enhancement layer" respectively is unclear. Accordingly, the relationship to the detailed explanation of the invention is unclear.

Likewise, "a first sub-sequence" and "a second sub-sequence" described in other claims are also unclear.

(2) Claim 1 describes "information indicating which video frames belong to the second sub-sequence", however, this description makes the invention unclear, for example, as to whether the information lists the frames which belong to the second sub-sequence among the all of the sequences, or indicates sub-sequences to which the respective video frames belong.

Moreover, the description "belong" make the invention unclear as to whether the information indicates that a video frame is a video frame of the second sub-sequence or is a video frame depending on the second sub-sequence.

As a result, the corresponding relationship to the

detailed explanation of the invention is unclear.

Likewise, the descriptions of claims 14 and 22 are unclear.

(3) Claim 4 describes "determining the dependencies of the video frames of the second sub-sequence on the basis of at least a scalability layer identifier and a sub-sequence identifier", however, this description could be understood to mean that the depended sub-sequence is indicated using the scalability layer identifier and the sub-sequence identifier and, therefore, the corresponding relationship to the detailed explanation of the invention is unclear.

(d) The description of the claims of this application does not satisfy the requirements prescribed in Article 36, paragraph 6, item 2 of the Patent Law on the following points.

NOTE

(1) Claim 1 describes "sub-sequence", however, the unit of sequence is unclear.

Likewise, the descriptions "sub-sequence" in other claims are also unclear.

(2) Claim 3 describes "coding the video sequence into a plural number of scalability layers", however, this description is unclear as to whether or not the video sequence is coded into coded data consisting of a plurality of scalability layers.

(3) Claim 3 describes "determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from a group comprising a video frame of a higher scalability layer, a video frame of another sub-sequence in the same scalability layer", however, this description is unclear as to whether prediction is performed dependently or information representing the dependency is formed in the video sequence.

(4) Claim 3 describes "one video frame of the second sub-sequence is predicted from a group comprising a video frame of a higher scalability layer, a video frame of another sub-sequence in the same scalability layer", however, from this description, the video frame to which a higher scalability layer or the same scalability layer is unclear.

(5) Claim 3 describes "a group comprising a video frame of a higher scalability layer, a video frame of another sub-sequence in the same scalability layer", however, from this description, the unit of the group, for example, whether the group is the entirety of the video sequence including the aforementioned video frame or consists only of the two aforementioned video frames is unclear.

(6) Claim 3 describes "predicted from a group", however, the description is unclear as to from which frame the prediction is performed.

(7) Claim 11 describes "at least some of the sub-sequence frames are temporally at least partly overlapping", however,

from this description, the layers whose frames overlap are unclear, for example, whether the frames of sub-sequence of different layer overlap, or the frames of sub-sequence of the same layer overlap.

Likewise, a description "partly temporally overlapping" in claim 20 is unclear.

(8) Claim 16 describes "comprise a number of image frames corresponding to the missing image numbers", however, the specific number represented by "a the number of" is unclear, for example, whether or not this description means image frames corresponding to the missing image numbers is unclear.

Likewise, the description "a number of" in claim 17 is also unclear.

(9) Claim 19 describes "determining an independently decodable sub-sequence", however, this description is unclear as to, for example, whether the sub-sequence is determined to be decodable, or the sub-sequence is identified by searching a decodable sub-sequence.

(10) Claim 19 describes "setting the value of the image number of the first video frame in the sub-sequence at zero", however, the description is unclear as to whether the setting is performed upon coding or decoding.

(11) Claim 20 describes "the last image frame in the decoding order", however, the description "last" is unclear as to whether the "last" is the last of entire sequence or the last of else.

(12) Claim 20 describes "group of said at least partly temporally overlapping image frames", however, this description is unclear as to whether the group means the entirety of the sequence including the temporally overlapped image frames, or a group of temporally overlapping sub-sequence having different layers, or else.

(13) Claim 22 describes "A computer program for coding ...(omitted)..., characterized in that the computer program comprises a program code for forming ...(omitted)...formed ...(omitted)... a program code for forming ...(omitted)... ; and a program code ...(omitted).", however, the subject which performs the coding is unclear, for example, whether or not the invention is directed to a program which directs the computer to perform the coding.

Moreover, the description "a program code ...(omitted)... and a program code" is unclear as to whether or not the program is used to perform each coding step.

Accordingly, the above description is unclear as to whether or not the program is used to direct the computer to perform each step described in claim 22.

Likewise, the description of claim 23 is unclear.

(14) Claim 24 is directed to "a video signal" and the category of the invention (a product invention or a process invention) for which patent is sought cannot be specified, thus, the invention is unclear.

(15) The invention claimed in each claim is unclear for the reason described in the Reason (c).

If an amendment is made, it should be noted that no new matter can be added, and, please assert in the argument that the amendments meet the requirements defined in the law, identifying the relevant descriptions of the original specification which support the amendments.

If another reason for rejection is found, a further notice of reasons for rejection will be issued.

List of Cited References

1. Japanese Unexamined Patent Publication (Kokai) No. 10-271511

Record of Results of Prior Art Search

- Searched Field : IPC
  H04B 7/24 - 7/68

This record of results of prior art search does not constitute reasons for rejection

Electronic Acknowledgement Receipt

EFS ID:	5179971
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	20-APR-2009
Filing Date:	25-JAN-2006
Time Stamp:	16:26:13
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Foreign Reference	JP10271511.pdf	1504119 30117798bbfb84b3a6032e8f31dcb02fa3fee47f	no	36

Warnings:	
Information:	

2	Transmittal Letter	IDSsupp6-TPO004C1.pdf	55965 2f253cd58e57415710d5c51ab6b5732b86a5e54f	no	2
Warnings:					
Information:					
3	Information Disclosure Statement (IDS) Filed (SB/08)	Form1449supp6-TPO004C1.pdf	55808 2685577bbd378ff04ecec4b9986ea482d0d83c97	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
4	NPL Documents	JP-OfficeAction-26032009.pdf	185000 3510bcc0224e2113b3825c248487950a40000166	no	6
Warnings:					
Information:					
5	NPL Documents	JP-OfficeActionEngligh-26032009pdf.pdf	189167 c8af3ce98c9773d06b6f6a1077e25746014439df	no	9
Warnings:					
Information:					
		Total Files Size (in bytes):	1990059		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平10－271511

(43)公開日　平成10年(1998)10月９日

(51)Int.Cl.[6]	識別記号	ＦＩ
Ｈ０４Ｎ　7/32		Ｈ０４Ｎ　7/137

審査請求　未請求　請求項の数９　ＯＬ　（全 36 頁）

(21)出願番号　特願平9－130614

(22)出願日　平成９年(1997)５月21日

(31)優先権主張番号　特願平9－9369

(32)優先日　平９(1997)１月22日

(33)優先権主張国　日本（ＪＰ）

(71)出願人　000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地

(72)発明者　畑　幸一
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(72)発明者　栄藤　稔
大阪府門真市大字門真1006番地　松下電器産業株式会社内

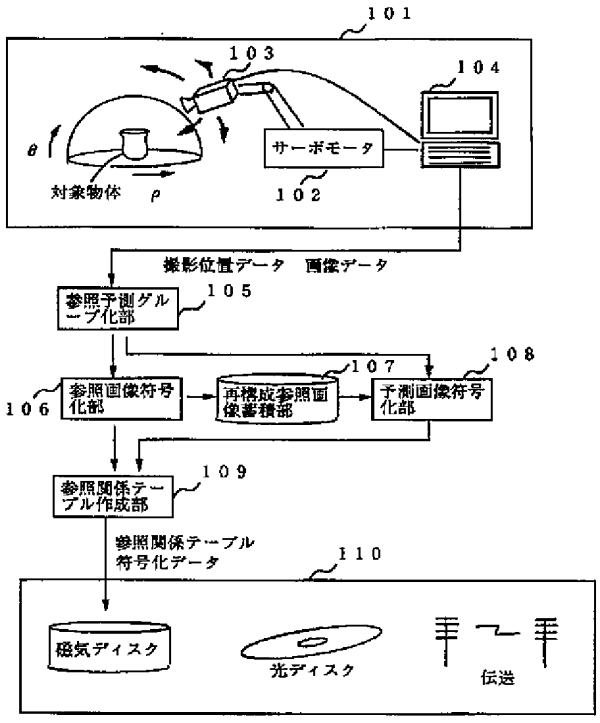
(74)代理人　弁理士　滝本　智之　（外１名）

(54)【発明の名称】　画像符号化装置と画像復号化装置

(57)【要約】

【課題】　被符号化画像群の符号化データに予測符号化の際の参照関係と撮影位置と符号化データ位置のテーブルを付加することで、効率のよく予測符号化でき、要求に効率よく対応できる復号化を実現する。

【解決手段】　複数の被符号化画像とその撮影位置データを入力とし、前記複数の被符号化画像を参照画像と参照画像から予測符号化される予測画像にグループ化する参照予測グループ化部１０５と、参照画像を符号化する参照画像符号化部１０６と、再構成参照画像を用いて予測画像を符号化する予測画像符号化部１０８と、参照関係と符号化データ位置と撮影位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データを出力する参照関係テーブル作成部１０９手段を備えたことを特徴とする。



1

【特許請求の範囲】

【請求項１】複数の被符号化画像とその撮影位置データを入力とし、前記複数の被符号化画像を参照画像と参照画像から予測符号化される予測画像にグループ化する参照予測グループ化手段と、参照画像を符号化する参照画像符号化手段と、再構成参照画像を用いて予測画像を符号化する予測画像符号化手段と、参照関係と符号化データ位置と撮影位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データを出力する参照関係テーブル作成手段を備えたことを特徴とする画像符号化装置。

【請求項２】再生すべき画像の要求と、画像が符号化された符号化データと、参照予測関係と撮影位置と符号化データ位置を記述した参照関係テーブルを入力とし、参照関係テーブルから要求される画像の復号化に必要な符号化データを選択する符号化データ選択手段と、符号化データを復号化し、再構成参照画像を出力する参照画像復号化手段と、符号化データを復号化し再構成予測画像を出力する予測画像復号化手段を備えたことを特徴とする画像復号化装置。

【請求項３】撮影位置と、参照画像符号化データ位置と予測画像符号化データ位置を記述したデータ位置テーブルと、予測画像符号化データについては、参照画像に対する変形情報とこの変形情報を用いて変形された画像に対する残差情報で構成されることを特徴とする符号化データを入力とし、再生すべき画像要求を逐次監視する画像要求監視手段と、画像要求と画像復号化状況とデータ位置テーブルと符号化データから、復号化する符号化データを選択する符号化データ選択手段と、符号化データから再構成参照画像を復号化する再構成参照画像復号化手段と、再構成参照画像と符号化データの変形情報から変形参照画像を復号化する変形参照画像復号化手段と、変形画像と符号化データの残差情報から再構成予測画像を復号化する再構成予測画像復号化手段を備え、再構成参照画像復号化手段と変形参照画像復号化手段および再構成予測画像復号化手段の出力を選択的に出力することを特徴とする画像復号化装置。

【請求項４】再生すべき画像の要求と、撮影位置と参照画像符号化データ位置と予測画像符号化データ位置を記述したデータ位置テーブルと、画像が符号化された符号化データを入力とし、復号化する符号化データを選択する符号化データ選択手段と、符号化データから参照画像を復号化する参照画像復号化手段と、再構成された参照画像を複数枚蓄積する再構成参照画像群蓄積手段と、符号化データから予測画像を復号化する予測画像復号化手段と、再構成された予測画像を複数枚蓄積する再構成予測画像群蓄積手段と、再生すべき画像の要求に応じて蓄積すべき画像の撮影位置を計算し、前記再構成参照画像群蓄積手段と再構成予測画像群蓄積手段に蓄積されている画像量を一定量以下に保つ画像蓄積検査手段を備え、

2

前記復号処理済みの蓄積されている画像から再生画像を出力することを特徴とする画像復号化装置。

【請求項５】撮影位置と参照画像符号化データ位置と予測画像符号化データ位置を記述したデータ位置テーブルと、符号化データを入力とし、再生すべき画像の要求を逐次監視し、画像要求を出力する画像要求監視手段と、現在の画像要求と前回の画像要求と既に復号化され蓄積された画像の情報から、新たに復号化する符号化データを選択する符号化データ選択手段と、撮影位置に関して現在の画像要求近傍の画像の符号化データを選択する予測符号化データ選択手段と、符号化データを参照画像に復号化する参照画像復号化手段と、符号化データを予測画像に復号化する予測画像復号化手段と、復号化された参照画像を蓄積する再構成参照画像群蓄積手段と、復号化された予測画像を蓄積する再構成予測画像群蓄積手段とを有し、画像要求を監視し再生する一方で、現在の画像要求近傍の画像の符号化データを選択し復号を行なうことを特徴とする画像復号化装置。

【請求項６】複数の被符号化画像と画像撮影位置データを入力とし、前記複数の被符号化画像を参照画像と、参照画像から予測される予測画像にグループ化する参照予測グループ化手段と、参照画像と予測画像の間の変形拘束式を推定する変形拘束式推定手段と、参照画像を符号化する参照画像符号化手段と、変形拘束式に基づいて変形を拘束して予測画像を符号化する拘束予測画像符号化手段、参照予測グループの参照関係と符号化データ位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データを出力する参照関係テーブル作成手段を備えたことを特徴とする画像符号化装置。

【請求項７】再生すべき画像の要求と、画像が符号化された符号化データと、参照予測関係と撮影位置と符号化データ位置を記述した参照関係テーブルを入力とし、参照関係テーブルから要求される画像の復号化に必要な符号化データを選択する符号化データ選択手段と、参照画像符号化データを復号化し、再構成参照画像を出力する参照画像復号化手段と、変形拘束式に基づいて変形を拘束して予測画像符号化データを復号化し再構成予測画像を出力する拘束予測画像復号化手段を備えたことを特徴とする画像復号化装置。

【請求項８】複数の被符号化画像とその撮影位置データと光源位置データを入力とし、前記複数の被符号化画像を参照画像と参照画像から予測符号化される予測画像にグループ化する参照予測グループ化手段と、参照画像を符号化する参照画像符号化手段と、再構成参照画像を用いて予測画像を符号化する予測画像符号化手段と、参照関係と符号化データ位置と撮影位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データを出力する参照関係テーブル作成手段を備えたことを特徴とする画像符号化装置。

【請求項９】再生すべき画像の要求と、画像が符号化さ

れた符号化データと、参照予測関係と撮影位置と光源位置と符号化データ位置を記述した参照関係テーブルを入力とし、参照関係テーブルから要求される画像の復号化に必要な符号化データを選択する符号化データ選択手段と、符号化データを復号化し、再構成参照画像を出力する参照画像復号化手段と、符号化データを復号化し再構成予測画像を出力する予測画像復号化手段を備えたことを特徴とする画像復号化装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は画像の伝送・蓄積のための画像符号化装置と画像復号化装置に関するものである。

【０００２】

【従来の技術】従来、３次元物体を計算機上で表現する際、物体の形状情報や表面の材質情報を計算機に入力し、見え方を計算し、シュミレーションする手法が取られているが、複雑な形状や細かな質感を表現するには、多くのデータを入力する必要があり、シュミレーションの計算量も膨大なものとなる。

【０００３】一方、より小さい計算量で３次元物体を表現するために、イメージベースドレンダリング（Image-BasedRendering）と呼ばれる、多くの視点からの画像をあらかじめ撮影・蓄積・伝送しておき、それらを要求に応じて表示する手法も取られている（「クイックタイムブイアール」（S. E. Chen、 "QuickTime VR — Animage-Based Approach to Virtual Environment Navigation"、 SIGGRAPH'90、 pp. 29—38、 1990））。

【０００４】

【発明が解決しようとする課題】Image-Based Rendering の手法では、多くの要求に応えるために、多くの視点からの画像をあらかじめ撮影しておくため、膨大なデータ量を蓄積・伝送しなければならない、これらのデータは画像として約１万枚、約２．４ギガバイトを越えることもある。したがって、これらの膨大なデータ量を、効率的に蓄積・伝送することが課題となっている。

【０００５】本発明は、従来のこのような課題を考慮し、蓄積・伝送すべき多数の画像の中から少数の代表画像を選び、各代表画像で複数枚の画像を表現し、その依存関係を付加することによって、効率的に複数の画像を符号化する画像符号化装置や、段階的な復号化や、次の要求の予測を行うことによって効率的に画像を符号化または復号化する方式とその装置を提供することを目的とする。

【０００６】

【課題を解決するための手段】第１の発明は、複数の被符号化画像とその撮影位置データを入力とし、前記複数の被符号化画像を参照画像と参照画像から予測符号化される予測画像にグループ化する参照予測グループ化手段と、参照画像を符号化する参照画像符号化手段と、再構

成参照画像を用いて予測画像を符号化する予測画像符号化手段と、参照関係と符号化データ位置と撮影位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データを出力する参照関係テーブル作成手段を備えたことを特徴とする画像符号化装置である。

【０００７】第２の発明は、再生すべき画像の要求と、画像が符号化された符号化データと、参照予測関係と撮影位置と符号化データ位置を記述した参照関係テーブルを入力とし、参照関係テーブルから要求される画像の復号化に必要な符号化データを選択する符号化データ選択手段と、符号化データを復号化し、再構成参照画像を出力する参照画像復号化手段と、符号化データを復号化し再構成予測画像を出力する予測画像復号化手段を備えたことを特徴とする画像復号化装置である。

【０００８】第３の発明は撮影位置と、参照画像符号化データ位置と予測画像符号化データ位置を記述したデータ位置テーブルと、予測画像符号化データについては、参照画像に対する変形情報とこの変形情報を用いて変形された画像に対する残差情報で構成されることを特徴とする符号化データを入力とし、再生すべき画像要求を逐次監視する画像要求監視手段と、画像要求と画像復号化状況とデータ位置テーブルと符号化データから、復号化する符号化データを選択する符号化データ選択手段と、符号化データから再構成参照画像を復号化する再構成参照画像復号化手段と、再構成参照画像と符号化データの変形情報から変形参照画像を復号化する変形参照画像復号化手段と、変形画像と符号化データの残差情報から再構成予測画像を復号化する再構成予測画像復号化手段を備え、再構成参照画像復号化手段と変形参照画像復号化手段および再構成予測画像復号化手段の出力を選択的に出力することを特徴とする画像復号化装置である。

【０００９】第４の発明は、再生すべき画像の要求と、撮影位置と参照画像符号化データ位置と予測画像符号化データ位置を記述したデータ位置テーブルと、画像が符号化された符号化データを入力とし、復号化する符号化データを選択する符号化データ選択手段と、符号化データから参照画像を復号化する参照画像復号化手段と、再構成された参照画像を複数枚蓄積する再構成参照画像群蓄積手段と、符号化データから予測画像を復号化する予測画像復号化手段と、再構成された予測画像を複数枚蓄積する再構成予測画像群蓄積手段と、再生すべき画像の要求に応じて蓄積すべき画像の撮影位置を計算し、前記再構成参照画像群蓄積手段と再構成予測画像群蓄積手段に蓄積されている画像量を一定量以下に保つ画像蓄積検査手段を備え、前記復号処理済みの蓄積されている画像から再生画像を出力することを特徴とする画像復号化装置である。

【００１０】第５の発明は、撮影位置と参照画像符号化データ位置と予測画像符号化データ位置を記述したデータ位置テーブルと、符号化データを入力とし、再生すべ

5

き画像の要求を逐次監視し、画像要求を出力する画像要求監視手段と、現在の画像要求と前回の画像要求と既に復号化され蓄積された画像の情報から、新たに復号化する符号化データを選択する符号化データ選択手段と、撮影位置に関して現在の画像要求近傍の画像の符号化データを選択する予測符号化データ選択手段と、符号化データを参照画像に復号化する参照画像復号化手段と、符号化データを予測画像に復号化する予測画像復号化手段と、復号化された参照画像を蓄積する再構成参照画像群蓄積手段と、復号化された予測画像を蓄積する再構成予測画像群蓄積手段とを有し、画像要求を監視し再生する一方で、現在の画像要求近傍の画像の符号化データを選択し復号を行なうことを特徴とする画像復号化装置である。

【００１１】第６の発明は、複数の被符号化画像と画像撮影位置データを入力とし、前記複数の被符号化画像を参照画像と、参照画像から予測される予測画像にグループ化し、参照画像と予測画像の間の変形拘束式を推定し、参照画像を符号化し、変形拘束式に基づいて変形を拘束して予測画像を符号化し、参照予測グループの参照関係と符号化データ位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データを出力することを特徴とする画像符号化装置である。

【００１２】第７の発明は、再生すべき画像の要求と、画像が符号化された符号化データと、参照予測関係と撮影位置と符号化データ位置を記述した参照関係テーブルを入力とし、参照関係テーブルから要求される画像の復号化に必要な符号化データを選択し、参照画像符号化データを復号化し、再構成参照画像を出力し、変形拘束式に基づいて変形を拘束して予測画像符号化データを復号化し再構成予測画像を出力することを特徴とする画像復号化装置である。

【００１３】第８の発明は、複数の被符号化画像とその撮影位置データと光源位置データを入力とし、前記複数の被符号化画像を参照画像と参照画像から予測符号化される予測画像にグループ化し、参照画像を符号化し、再構成参照画像を用いて予測画像を符号化し、参照関係と符号化データ位置と撮影位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データを出力することを特徴とする画像符号化装置である。

【００１４】第９の発明は、再生すべき画像の要求と、画像が符号化された符号化データと、参照予測関係と撮影位置と光源位置と符号化データ位置を記述した参照関係テーブルを入力とし、参照関係テーブルから要求される画像の復号化に必要な符号化データを選択し、符号化データを復号化し再構成参照画像を出力し、符号化データを復号化し再構成予測画像を出力することを特徴とする画像復号化装置である。

【００１５】第１の発明に基づく画像符号化装置では、参照予測グループ化手段が、入力である複数の被符号化

6

画像とその撮影位置データから被符号化画像を参照画像と参照画像から予測符号化される予測画像からなる複数のグループに分ける。参照画像は参照画像符号化手段により符号化され、その再構成画像を用いて予測画像符号化手段が予測画像を符号化する。そして、参照関係テーブル作成手段が、参照画像に対する予測画像の参照関係、符号化データ位置、撮影位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データを出力する。

【００１６】第２の発明に基づく画像復号化装置では、再生すべき画像の要求、画像符号化データおよび参照予測関係、撮影位置、符号化データ位置を記述した参照関係テーブルを入力とする。符号化データ選択手段は、要求される画像の復号化に必要な符号化データを参照関係テーブルに基づいて選択する。

【００１７】選択された符号化データが参照画像であれば、参照画像復号化手段により復号化され、再構成参照画像として出力される。また選択された符号化データが予測画像データであれば、予測画像復号化手段により復号化され再構成予測画像として出力される。

【００１８】第３の発明に基づく画像復号化装置では、撮影位置と参照画像符号化データ位置と予測画像符号化データ位置を記述したデータ位置テーブルと、予測画像符号化データについては、参照画像に対する変形情報とこの変形情報を用いて変形された画像に対する残差情報で構成されることを特徴とする符号化データを入力とする。画像要求監視手段は再生すべき画像要求を逐次監視する。符号化データ選択手段は、画像要求と画像復号化状況に応じてデータ位置テーブルと符号化データから、復号化すべき符号化データを選択する。

【００１９】選択された符号化データが参照画像データであれば、再構成参照画像復号化手段がその符号化データから再構成参照画像を復号化する。また選択された符号化データが予測画像データであれば、変形参照画像復号化手段または再構成予測画像復号化手段が予測画像データを復号する。変形参照画像復号化手段によれば、再構成参照画像と符号化データの変形情報から変形参照画像が復元され、再構成予測画像復号化手段によれば、変形画像と符号化データの残差情報から再構成予測画像が復元される。再構成参照画像復号化手段と変形参照画像復号化手段および再構成予測画像復号化手段の出力は、画像要求と画像復号化状況に応じてデータ位置テーブルにより選択的に出力されることになる。

【００２０】第４の発明に基づく画像復号化装置では、再生すべき画像の要求と、撮影位置と参照画像符号化データ位置と予測画像符号化データ位置を記述したデータ位置テーブルと、画像が符号化された符号化データを入力とする。

【００２１】符号化データ選択手段は再生すべき画像の要求を入力として復号化べき符号化データを選択する。

選択された符号化データが参照画像データであれば、参照画像復号化手段が、選択された符号化データが予測画像データであれば、予測画像復号化手段が画像の再構成を行ない、その結果はそれぞれ再構成参照画像群蓄積手段と再構成予測画像群蓄積手段に蓄積される。この時、画像蓄積検査手段が、再生すべき画像の要求に応じて蓄積すべき画像の撮影位置を計算し、前記再構成参照画像群蓄積手段と再構成予測画像群蓄積手段に蓄積されている画像量を一定量以下に保つよう廃棄する蓄積画像を決定する。出力画像は復号処理済みの蓄積されている画像から選択されて出力される。

【００２２】第５の発明に基づく画像復号化装置では、撮影位置と参照画像符号化データ位置と予測画像符号化データ位置を記述したデータ位置テーブルと、符号化データを入力とする。画像要求監視手段は、再生すべき画像の要求を逐次監視し、画像要求を出力する。符号化データ選択手段は、現在の画像要求と前回の画像要求と既に復号化され蓄積された画像の情報から、新たに復号化する符号化データを選択する。その一方で、予測符号化データ選択手段は、撮影位置に関して現在の画像要求近傍の画像の符号化データを選択する。以上の二つの選択手段で選ばれた符号化データは選択された符号化データが参照画像データであれば、参照画像復号化手段が、選択された符号化データが予測画像データであれば、予測画像復号化手段が画像の再構成を行ない、その結果はそれぞれ再構成参照画像群蓄積手段と再構成予測画像群蓄積手段に蓄積される。

【００２３】このように、画像要求を監視し再生する一方で、現在の画像要求近傍の画像の符号化データを選択し復号を行ない、画像を出力する。

【００２４】第６の発明に基づく画像符号化装置は、複数の被符号化画像と画像撮影位置データを入力とする。参照予測グループ化手段では、前記複数の被符号化画像を、参照画像と、参照画像から予測される予測画像にグループ化する。変形拘束式推定手段では、参照画像と予測画像の間の変形拘束式を推定する。

【００２５】参照画像符号化手段では、参照画像を符号化する。拘束予測画像符号化手段では、変形拘束式に基づいて変形を拘束して予測画像を符号化する。参照関係テーブル作成手段では、参照予測グループの参照関係と符号化データ位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データを出力する。

【００２６】第７の発明に基づく画像復号化装置では、再生すべき画像の要求と、画像が符号化された符号化データと、参照予測関係と撮影位置と符号化データ位置を記述した参照関係テーブルを入力とする。符号化データ選択手段では、参照関係テーブルから要求される画像の復号化に必要な符号化データを選択する。参照画像復号化手段では、参照画像符号化データを復号化し、再構成参照画像を出力する。拘束予測画像復号化手段では、変

形拘束式に基づいて変形を拘束して予測画像符号化データを復号化し再構成予測画像を出力する。

【００２７】第８の発明に基づく画像符号化装置では、複数の被符号化画像とその撮影位置データと光源位置データを入力とする。参照予測グループ化手段では、前記複数の被符号化画像を参照画像と参照画像から予測符号化される予測画像にグループ化する。参照画像符号化手段では、参照画像を符号化する。予測画像符号化手段では、再構成参照画像を用いて予測画像を符号化する。参照関係テーブル作成手段では、参照関係と符号化データ位置と撮影位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データを出力する。

【００２８】第９の発明に基づく画像復号化装置では、再生すべき画像の要求と、画像が符号化された符号化データと、参照予測関係と撮影位置と光源位置と符号化データ位置を記述した参照関係テーブルを入力とする。

【００２９】符号化データ選択手段では、参照関係テーブルから要求される画像の復号化に必要な符号化データを選択する。参照画像復号化手段では、符号化データを復号化し、再構成参照画像を出力する。予測画像復号化手段では、符号化データを復号化し再構成予測画像を出力する。

【００３０】

【発明の実施の形態】以下、本発明にかかる実施の形態について図面を参照しながら説明する。

【００３１】図１は本発明の第１の実施の形態である画像符号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【００３２】同図において、撮影位置制御型画像撮像部（１０１）は、対象物体の画像を撮影し、その撮影位置データと画像データを出力する。参照予測グループ部（１０５）は、撮影位置データと画像データを参照画像と一枚以上の予測画像からなる参照予測グループにグループ化し、参照画像を参照画像符号化部（１０６）に出力し、予測画像を予測画像符号化部（１０８）に出力する。

【００３３】参照画像符号化部（１０６）は、参照画像を符号化し、再構成参照画像を作成し、符号化データを参照関係作成部（１０９）に出力し、再構成参照画像を再構成参照画像蓄積部（１０７）に出力する。再構成参照画像蓄積部（１０７）は、再構成参照画像を蓄積し、予測画像符号化部（１０８）に再構成参照画像を出力する。

【００３４】予測画像符号化部（１０８）は、予測画像を符号化し、符号化データを参照関係作成部（１０９）に出力する。参照関係テーブル作成部（１０９）は、符号化データから参照関係と符号化データ位置と撮影位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データをデータ記録・伝送部（１１０）に出力する。データ記録・伝送部（１１０）は、用途に応じ

て記録・伝送する。

【００３５】以上のように構成された本実施の形態の画像符号化装置の動作を説明する。まず、撮影位置制御型画像撮像部（１０１）では、対象物体の周囲を網羅するように、異なる視点から画像を撮影する。計算機によってサーボモータ（１０２）を対象物体の正面から水平方向にρ度、水平面から垂直方向にθ度の位置にカメラ（１０３）を移動させ、撮影位置・画像データを計算機（１０４）によって取得する。画像データは、クロマキーによって対象物体を背景と分離し、透過度情報を示すアルファプレーンを付加し、輝度プレーン（２０１）、色差１プレーン（２０２）、色差２プレーン（２０３）、アルファプレーン（２０４）で構成する。

【００３６】ここで、ρ，θは、

【００３７】

【数１】

$$0 \leq \rho < 360,\ 0 \leq \theta \leq 90,\ (\rho,\theta\text{は整数})$$

【００３８】とする。このように取得された撮影位置・画像データは、図３に示すように、各画像データの先頭に水平方向撮影角度ρと、垂直方向撮影角度θを付加した書式で出力される。

【００３９】参照予測グループ化部（１０５）では、参照画像（４０１）によって類似した画像を予測画像（４０２）として予測符号化するために、$\rho-\theta$空間において＊て等間隔に参照画像を配置し、参照画像の近傍の画像をその参照画像より予測される予測画像にグループ化するため、撮影位置Ｐ＝（ρ，θ）を用いて、以下のように画像それぞれに参照画像か予測画像のフラグｆ（Ｐ）を割り当てる。

【００４０】

【数２】

$$f(\mathbf{P}) = \begin{cases} 0 & \mathbf{P}=(Sn,Sm)\ (n,m\text{は整数}) \\ 1 & \text{それ以外} \end{cases}$$

【００４１】さらに、以下のようにインデックスｎ（Ｐ）を割り当てる。

【００４２】

【数３】

$$n(\mathbf{P}) = \lceil \tfrac{359}{S} \rceil \times n + m,\ n,m = \min_{n,m} d(\mathbf{P},\mathbf{Q})$$

【００４３】以上において、Ｑ＝（Ｓｎ，Ｓｍ）である。Ｓは参照画像間の間隔でありＳ＝５である。「・」は、・を越えない整数を表わす。また、水平方向撮影位置で折り返すため、

【００４４】

【数４】

$$\mathbf{P}_1 = (\rho_1,\theta_1),\ \mathbf{P}_2 = (\rho_2,\theta_2)$$

【００４５】とすると、

【００４６】

【数５】

$$d(\mathbf{P}_1,\mathbf{P}_2) = \min((\rho_1-\rho_2)^2,(\rho_1-\rho_2-360)^2,(\rho_1-\rho_2+360)^2) + (\theta_1-\theta_2)^2$$

【００４７】である。これらを、インデックス順に図５の示す書式で、参照画像は参照画像符号化部（１０６）へ、予測画像は予測画像符号化部（１０７）へ出力する。

【００４８】参照画像符号化部（１０６）では、参照画像をその画像データのみで符号化もしくは、その符号化データのみで復号化出来るよう復号化しておく、アルファプレーンを含んだ、そのような符号化する方法には、ＭＰＥＧ４（MPEG Video Group、"MPEG-4 Video Verification Model Version 5.0"、ISO/IEC JTC1/SC29/WG11 Document N1469、 1996）で標準化が進められている動画像符号化方法のＩ－ＰＩＣＴＵＲＥ符号化方式がある。

【００４９】この方法を用いて、参照画像符号化部（１０６）で参照画像を符号化し、符号化データを参照関係テーブル作成部（１０９）に出力し、参照画像の再構成画像を作成し、再構成参照画像蓄積部（１０７）に出力する。図６に参照画像符号化部（１０６）のフローチャートを示す。

【００５０】再構成参照画像蓄積部（１０７）では、再構成参照画像を蓄積しておき、予測画像符号化部（１０８）に出力する。参照予測グループ化部（１０５）において、図５のように参照画像と予測画像が整列されて出力されるので、再構成参照画像蓄積部（１０８）は、画像１枚分の容量を持っていれば十分である。

【００５１】予測画像符号化部（１０８）では、予測画像を再構成参照画像を用いて符号化する。参照画像と同様に、アルファプレーンを含んだ予測画像符号化方法には、ＭＰＥＧ４による動画像符号化のＰ－ＰＩＣＴＵＲＥ符号化方式がある。この方法を用いて予測画像を符号化し、符号化データを参照関係テーブル作成部（１０９）に出力する。図７に参照画像符号化部（１０６）のフローチャートを示す。

【００５２】参照関係テーブル作成部（１０９）では、各符号化部より出力された図８のようなデータを順に走査し、図９のような参照関係テーブルと図１０のような符号化データを作成し、データ記録・伝送部（１１０）に出力する。

【００５３】データ記録・伝送部（１１０）では、参照関係テーブルと符号化データを磁気ディスクや光ディスクに蓄積しておいたり、ネットワークなどで伝送する。

【００５４】以上のように、本実施の形態では、異なる多くの視点から撮影された多視点画像を、撮影位置データが近傍の類似した画像である参照画像と１枚以上の予測画像にグループ化し、効率よく符号化することが可能である。さらに、画像を各参照予測グループで参照画像が先頭になるように整列して、符号化手段に出力することで、計算量と計算に必要な蓄積容量を少なくしている。また、その整列順に符号化データ位置を格納したテーブルを付加することで、ランダムアクセスに対応した

効率的な復号化が可能である。

【００５５】図１１は本発明の第２の実施の形態である画像復号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【００５６】同図において、データ記憶・伝送部（１１０１）では、参照関係テーブルと符号化データを蓄積・伝送し、参照テーブルと符号化データを符号化データ選択部（１１０２）に出力する。符号化データ選択部（１１０２）では、参照関係テーブルからデータの位置を検索し、符号化データから必要なデータを参照画像復号化部（１１０３）や予測画像復号化部（１１０５）に出力する。参照画像復号化部（１１０３）では、参照画像符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（１１０４）に出力し、必要なら画像表示手段（１１０６）にも出力する。

【００５７】再構成参照画像蓄積部（１１０５）は、再構成参照画像を蓄積し、予測画像復号化部（１１０５）に出力する。予測画像復号化部（１１０５）では、予測画像符号化データを復号化し、再構成予測画像を画像表示部（１００６）に出力する。画像表示部（１１０６）は入力された画像を蓄積しておき、要求により表示する。

【００５８】以上のように構成された本実施の形態の画像復号化装置の動作を説明する。データ記憶・伝送部（１１０１）では、参照関係テーブルと符号化データを蓄積・伝送し、参照テーブルと符号化データを符号化データ選択部（１１０２）に出力する。

【００５９】符号化データ選択部（１１０２）では、参照関係テーブルから符号化データ位置を検索し、データ記憶・伝送部（１１０１）から、符号化データ位置の符号化データを取得する。参照／予測フラグと参照符号化データを、参照画像復号化部（１１０３）に出力し、予測符号化データがあれば、予測画像復号化部（１１０５）に出力する。但し、参照画像復号化部（１１０３）に出力する参照符号化データが前回と同じ場合は、参照画像はすでに復号化され、蓄積されているので、参照画像復号化部（１１０３）への出力は省略する。参照画像復号化部（１１０３）では、参照画像が符号化された参照符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（１１０４）に出力する。

【００６０】また、参照／予測フラグにおいて、参照画像であれば、画像表示部（１００６）に再構成参照画像を出力する。再構成参照画像蓄積部（１１０４）では、再構成された参照画像を蓄積し、予測画像復号化部（１１０５）に出力する。

【００６１】予測画像復号化部（１１０５）では、予測画像が符号化された予測符号化データを再構成参照画像を用いて復号化し、画像表示部（１１０６）に再構成予測画像を出力する。

【００６２】画像表示部（１１０６）では、復号化された画像を蓄積しておき、計算機に接続されたマウスにより検索し、表示する。

【００６３】以上のように、本実施の形態では、多くの異なる視点から撮影した画像を、撮影位置が近傍の画像をグループ化し、グループを予測符号化された符号化データを、その参照予測関係と符号化データ位置と撮影位置を記述したテーブルを用いることにより、効率的に復号化することが可能である。

【００６４】また、参照関係テーブルはグループ順に記述されているので、同一参照予測グループ内が連続的に復号化されるので、再構成参照画像を蓄積しておき、予測符号化データを復号化するだけで、効率的に画像を復号化することが可能である。

【００６５】図１２は本発明の第３の実施の形態である画像復号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【００６６】同図において、データ記憶・伝送部（１２０１）では、参照関係テーブルと符号化データを蓄積・伝送し、参照テーブルをランダムアクセステーブル作成部（１２０２）に出力し、符号化データを符号化データ選択部（１２０３）に出力する。画像要求部（１２０４）では、要求する画像の撮影位置を符号化データ選択部（１２０３）に出力する。ランダムアクセステーブル作成部（１２０２）では、撮影位置から符号化データに効率的にランダムアクセスできるように、テーブルを変更する。

【００６７】符号化データ選択部（１２０３）では、画像要求に基づいて、ランダムアクセステーブルから、必要なデータの位置を検索し、符号化データから必要なデータを参照画像復号化部（１２０５）や予測画像復号化部（１２０７）に出力する。参照画像復号化部（１２０５）では、参照画像符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（１２０６）に出力し、必要なら画像表示部（１２０８）にも出力する。

【００６８】再構成参照画像蓄積部（１２０６）は、再構成参照画像を蓄積し、予測画像復号化部（１２０７）に出力する。予測画像復号化部（１２０７）では、予測画像符号化データを復号化し、再構成予測画像を画像表示部（１２０７）に出力する。画像表示部（１２０８）は入力された画像を表示する。

【００６９】以上のように構成された本実施の形態の画像復号化装置の動作を説明する。データ記憶・伝送部（１２０１）では、参照関係テーブルと符号化データを蓄積・伝送し、参照テーブルをランダムアクセステーブル作成部（１２０２）に出力する。また、符号化データ選択部（１２０３）の要求により、符号化データを符号化データ選択部（１２０３）に出力する。

【００７０】画像要求部（１２０４）では、計算機に接続されたマウスを操作することによって、ユーザが見たい視点位置を指定し、その視点位置に対応する撮影位置

を符号化データ選択部（１２０３）に出力する。

【００７１】ランダムアクセステーブル作成部（１２０２）では、図９に示す参照関係テーブルを、撮影位置から符号化データ位置を効率的に検索できるよう、ランダムアクセステーブルを作成する。図９中の参照ヘッダと予測ヘッダは、図１３のようなランダムアクセスヘッダ（Ｎバイト）に変換され、撮影位置（ρ，θ）のランダムアクセスヘッダはランダムアクセステーブルの先頭から、

【００７２】

【数６】

$$(359\rho+\theta)N$$

【００７３】の位置に配置し、図１４に示すランダムアクセステーブルを作成する。ただし、参照画像の場合、予測符号化データ位置の欄は空欄にしておく。

【００７４】符号化データ選択部（１２０３）では、要求画像の撮影位置データ（α，β）を用いて、ランダムアクセステーブルの先頭から、

【００７５】

【数７】

$$(359\alpha+\beta)N$$

【００７６】の位置のランダムアクセスヘッダを検索し、データ記憶・伝送手段（１２０１）から、符号化データ位置の符号化データを取得する。参照／予測フラグと参照符号化データは、参照画像復号化部（１２０５）に出力し、予測符号化データがあれば、予測画像復号化部（１２０７）に出力する。

【００７７】ただし、参照画像復号化部（１２０５）に出力する参照符号化データが前回と同じ場合は、参照画像はすでに復号化され、蓄積されているので、参照画像復号化部（１２０５）への出力は省略する。

【００７８】参照画像復号化部（１２０５）では、参照画像が符号化された参照符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（１２０６）に出力する。また、参照／予測フラグにおいて、参照画像であれば、画像表示部（１２０８）に再構成参照画像を出力する。

【００７９】再構成参照画像蓄積部（１２０６）では、再構成された参照画像を蓄積し、予測画像復号化部（１２０７）に出力する。

【００８０】予測画像復号化部（１１０７）では、予測画像が符号化された予測符号化データを再構成参照画像を用いて復号化し、画像表示部（１２０８）に再構成予測画像を出力する。画像表示部（１２０８）では、入力された画像を表示する。

【００８１】以上のように、本実施の形態では、多くの異なる視点から撮影した画像を、撮影位置が近傍の画像をグループ化し、効率的に予測符号化し、その参照予測関係と符号化データ位置と撮影位置を記述したテーブルを付加することにより、任意の撮影位置の画像を効率的に検索、復号化、表示することが可能である。また、連続的に要求される画像は、撮影位置が近傍のものが多く、同一参照予測グループ内である場合が多く、再構成参照画像を蓄積しておくことで、予測符号化データを復号化するだけで、効率的に画像を復号化、表示することが可能である。

【００８２】図１５は本発明の第４の実施の形態である画像符号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【００８３】同図において、撮影位置制御型画像撮像部（１５０１）は、対象物体の画像を撮影し、その撮影位置データと画像データを出力する。参照予測グループ部（１５０２）は、撮影位置データと画像データを画像データを参照画像と一枚以上の予測画像からなる参照予測グループにグループ化し、参照画像を参照画像符号化部（１５０３）に出力し、予測画像を予測画像符号化部（１５０５）に出力する。参照画像符号化部（１５０３）は、参照画像を符号化し、再構成参照画像を作成し、符号化データを参照関係テーブル作成部（１５０６）に出力し、再構成参照画像を再構成参照画像蓄積部（１５０４）に出力する。再構成参照画像蓄積部（１５０４）は、再構成参照画像を蓄積し、予測画像符号化部（１５０５）に再構成参照画像を出力する。予測画像符号化部（１５０５）は、予測画像を符号化し、符号化データを参照関係テーブル作成部（１５０６）に出力する。参照関係テーブル作成部（１５０６）は、符号化データから参照関係と符号化データ位置と撮影位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データをランダムアクセステーブル作成部（１５０７）に出力する。

【００８４】ランダムアクセステーブル作成部（１５０７）では、撮影位置から符号化データに効率的にランダムアクセスできるように、テーブルを変更し、ランダムアクセステーブルと符号化データをデータ記録・伝送部（１５０８）に出力する。データ記録・伝送部（１５０８）は、用途に応じて記録・伝送する。

【００８５】以上のように構成された本実施の形態の画像符号化装置の動作を説明する。まず、画像位置制御型画像撮像部（１５０１）は、画像位置制御型画像撮影部（１０１）と同様である。参照予測グループ化部（１５０２）は、参照予測グループ化部（１０５）と同様である。参照画像符号化部（１５０３）は、参照画像符号化部（１０６）と同様である。再構成参照画像蓄積部（１５０４）は、再構成参照画像蓄積部（１０７）と同様である。予測画像符号化部（１５０５）は、予測画像符号化部（１０８）と同様である。参照関係テーブル作成部（１５０６）は、参照関係テーブル作成部（１０９）と同様である。

【００８６】ランダムアクセステーブル作成部（１５０７）は、ランダムアクセステーブル作成部（１２０２）

と同様に、ランダムアクセステーブルを作成し、データ記録・伝送部（１１０）に符号化データとランダムアクセステーブルを出力する。

【００８７】データ記録・伝送部（１１０）では、ランダムアクセステーブルと符号化データを磁気ディスクや光ディスクに蓄積しておいたり、ネットワークなどで伝送する。

【００８８】以上のように、本実施の形態では、異なる多くの視点から撮影された多視点画像を、撮影位置データが近傍の類似した画像である参照画像と１枚以上の予測画像にグループ化し、効率よく符号化することが可能である。さらに、画像を各参照予測グループで参照画像が先頭になるように整列して、符号化手段に出力することで、計算量と計算に必要な蓄積容量を少なくしている。また、その撮影位置順に符号化データ位置を格納したテーブルを付加することで、ランダムアクセスに対応した効率的な復号化が可能である。

【００８９】図１６は本発明の第５の実施の形態である画像復号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【００９０】同図において、データ記憶・伝送部（１６０１）では、ランダムアクセステーブルと符号化データを蓄積・伝送し、ランダムアクセステーブルと符号化データを符号化データ選択部（１６０２）に出力する。画像要求部（１６０３）では、要求する画像の撮影位置を符号化データ選択部（１６０２）に出力する。

【００９１】符号化データ選択部（１６０２）では、画像要求に基づいて、ランダムアクセステーブルから、必要なデータの位置を検索し、符号化データから必要なデータを参照画像復号化部（１６０４）や予測画像復号化部（１６０６）に出力する。参照画像復号化部（１６０４）では、参照画像符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（１６０５）に出力し、必要なら画像表示部（１６０７）にも出力する。

【００９２】再構成参照画像蓄積部（１６０５）は、再構成参照画像を蓄積し、予測画像復号化部（１６０６）に出力する。予測画像復号化部（１６０６）では、予測画像符号化データを復号化し、再構成予測画像を画像表示部（１６０７）に出力する。画像表示部（１６０７）は入力された画像を表示する。

【００９３】以上のように構成された本実施の形態の画像復号化装置の動作を説明する。データ記憶・伝送部（１６０１）では、ランダムアクセステーブルと符号化データを蓄積・伝送し、ランダムアクセステーブルを符号化データ選択部（１６０２）に出力する。また、符号化データ選択部（１６０２）の要求により、符号化データを符号化データ選択部（１６０２）に出力する。

【００９４】画像要求部（１６０３）では、計算機に接続されたマウスを操作することによって、ユーザが見たい視点位置を指定し、その視点位置に対応する撮影位置を符号化データ選択部（１６０２）に出力する。符号化データ選択部（１６０２）は、符号化データ選択部（１２０３）と同様である。

【００９５】参照画像復号化部（１６０４）では、参照画像が符号化された参照符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（１６０５）に出力する。また、参照／予測フラグにおいて、参照画像であれば、画像表示部（１６０７）に再構成参照画像を出力する。再構成参照画像蓄積部（１６０５）では、再構成された参照画像を蓄積し、予測画像復号化部（１６０６）に出力する。

【００９６】予測画像復号化部（１６０６）では、予測画像が符号化された予測符号化データを再構成参照画像を用いて復号化し、画像表示部（１６０７）に再構成予測画像を出力する。画像表示部（１６０７）では、入力された画像を表示する。

【００９７】以上のように、本実施の形態では、多くの異なる視点から撮影した画像を、撮影位置が近傍の画像をグループ化し、効率的に予測符号化し、その参照予測関係と符号化データ位置と撮影位置を記述したテーブルを付加することにより、任意の撮影位置の画像を効率的に検索、復号化、表示することが可能である。また、連続的に要求される画像は、撮影位置が近傍のものが多く、同一参照予測グループ内である場合が多く、再構成参照画像を蓄積しておくことで、予測符号化データを復号化するだけで、効率的に画像を復号化、表示することが可能である。

【００９８】図１７は本発明の第６の実施の形態である画像復号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【００９９】同図において、データ記憶・伝送部（１７０１）では、ランダムアクセステーブルと符号化データを蓄積・伝送し、ランダムアクセステーブルと符号化データを符号化データ選択部（１７０５）に出力する。画像要求監視部（１７０２）は、画像の要求を監視し画像要求蓄積部（１７０３）に出力する。

【０１００】画像要求蓄積部（１７０３）は、画像要求を蓄積する。復号化状況蓄積部（１７０４）は復号化の状況を蓄積する。符号化データ選択部（１７０５）は、復号化状況と画像要求とランダムアクセステーブルと符号化データから復号化すべき符号化データを選択し、参照画像復号化部（１７０６）や変形画像復号化部（１７０８）や予測画像復号化部（１７１０）や出力画像選択部（１７１２）に出力する。参照画像復号化部（１７０６）は、参照画像符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（１７０７）に出力する。

【０１０１】再構成参照画像蓄積部（１７０７）は再構成参照画像を蓄積する。変形参照画像復号化部（１７０８）は符号化データの動きベクトル部分と再構成参照画像から、変形参照画像を復号化し、変形参照画像蓄積部

（１７０９）に出力する。

【０１０２】変形参照画像蓄積部（１７０９）は、変形参照画像を蓄積する。予測画像復号化部（１７１０）は、符号化データの残差成分部分と変形参照画像から、予測画像を復号化し、予測画像蓄積部（１７１１）に出力する。予測画像蓄積部（１７１１）は、予測画像を蓄積する。出力画像選択部（１７１２）は、画像を選択し、画像表示部（１７１３）に出力する。画像表示部（１７１３）は、画像を表示する。

【０１０３】以上のように構成された本実施の形態の画像復号化装置の動作を説明する。まず、データ記憶・伝送部（１７０１）では、ランダムアクセステーブルと符号化データを蓄積・伝送し、ランダムアクセステーブルを符号化データ選択部（１７０５）に出力する。また、符号化データ選択部（１７０５）により、符号化データが読み出される。

【０１０４】画像要求監視部（１７０２）は、マウスを持った計算機であり、ユーザがマウスを使ってする要求を常に監視し、画像要求蓄積部（１７０３）に蓄積する。画像要求蓄積部（１７０３）は、要求される画像の撮影位置を蓄積し、符号化データ選択部（１７０５）により読み出される。

【０１０５】復号化状況蓄積部（１７０４）では、復号化状況テーブル（１８０１）を蓄積し、符号化データ選択部（１７０４）により更新され、読み出される。復号化状況テーブル（１８０１）において、撮影位置は（ρ，θ）であり、復号化レベルは復号化の状況を示す。復号化レベルの意味を復号化レベルテーブル（１８０２）に示す。

【０１０６】符号化データ選択部（１７０５）は、要求する撮影位置とランダムアクセステーブルと復号化状況テーブルから、出力データと出力先と復号化状況テーブル更新方法を決定する。図１９に要求する撮影位置とランダムアクセステーブルと復号化状況テーブルの状態を列挙した状態テーブルを示す。ここで、Ｐは要求された画像の撮影位置、Ｑは要求された画像が予測画像の場合、その画像の復号化に必要な参照画像の撮影位置、Ｒは復号化状況テーブル（１８０１）に登録されている参照画像撮影位置、Ｓは復号化状況テーブル（１８０１）に登録されている予測画像撮影位置、Ｌは復号化状況テーブル（１８０１）に登録されている復号化レベルを表わす。

【０１０７】図２０に出力データ書式（２００１）と各状態に対応する出力データと出力先と更新後の復号化状況テーブルを列挙した出力データテーブル（２００２）を示す。出力データ書式（２００１）において出力画像フラグは最終的に出力されるべき画像を示すフラグであり、０は出力なし、１は再構成参照画像を出力、２は変形参照画像を出力、３は再構成予測画像を出力を示す。符号化データはランダムアクセステーブルより撮影位置に対応する符号化データ位置を検索し、データ記憶・伝送部（１７０１）より読み出す。

【０１０８】参照画像復号化部（１７０６）では、入力された出力画像フラグと符号化データを出力画像フラグは出力画像選択部（１７１２）に出力し、符号化データは参照画像に復号化し、再構成参照画像蓄積部（１７０７）に出力する。再構成参照画像蓄積部（１７０７）は、再構成参照画像を蓄積する。変形画像復号化部（１７０８）では、入力された出力画像フラグと符号化データを出力画像フラグは出力画像選択部（１７１２）に出力し、符号化データは符号化データの動きベクトル部分を復号化し、再構成参照画像を変形させて変形参照画像を作成し、変形参照画像蓄積部（１７０９）に出力する。変形参照画像蓄積部（１７０９）は、変形参照画像を蓄積する。

【０１０９】予測画像復号化部（１７１０）では、入力された出力画像フラグと符号化データを、出力画像フラグは出力画像選択部（１７１２）に出力し、符号化データは符号化データの残差成分部分を復号化し、変形参照画像に残差成分を加えて予測画像を作成し、再構成予測画像蓄積部（１７１１）に出力する。再構成予測画像蓄積部（１７１１）は、再構成予測画像を蓄積する。

【０１１０】出力画像選択部（１７１２）は、入力された出力画像フラグが、０ならば出力しない、１ならば再構成参照画像蓄積部（１７０６）より再構成参照画像を読み出し、画像表示部（１７１３）に出力する、２ならば変形参照画像蓄積部（１７０８）より変形参照画像を読み出し、画像表示部（１７１３）に出力する、３ならば再構成予測画像蓄積部（１７１１）より再構成予測画像を読み出し、画像表示部（１７１３）に出力する。画像表示部（１７１３）は、入力された画像を表示する。

【０１１１】以上のように、本実施の形態では、画像要求を監視し、蓄積する部と、復号化処理途中の変形画像を出力する部を備えており、画像要求の変化の速度と画像復号化処理の速度に対応して、自動的に変形画像か再構成予測画像を出力することにより、画像要求の変化が速いときや画像復号化処理の速度が遅いときは画質の良くない変形画像が、その逆のときには画質のよい再構成予測画像を表示することが可能である。

【０１１２】図２１は本発明の第７の実施の形態である画像復号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【０１１３】同図において、データ記憶・伝送部（２１０１）では、ランダムアクセステーブルと符号化データを蓄積・伝送し、ランダムアクセステーブルと符号化データを符号化データ選択部（２１０４）に出力する。画像要求部（２１０２）は、要求する画像の撮影位置を画像蓄積検査部（２１０３）に出力する。画像蓄積検査部（２１０３）では、再構成参照画像群蓄積部（２１０８）と再構成予測画像群蓄積部（２１０９）に蓄積され

ている画像の重要度を撮影位置に基づいて計算し、最大蓄積量を越えないように重要度の低い画像を消去する。符号化データ選択部（２１０４）では、蓄積された画像とランダムアクセステーブルに基づいて必要な符号化データを選択し、参照画像復号化部（２１０５）や予測画像復号化部（２１０６）や参照予測画像復号化部（２１０７）に出力する。

【０１１４】再構成参照画像群蓄積部（２１０８）では、複数枚の再構成参照画像と各画像の撮影位置を蓄積し、出力画像選択部（２１１０）に出力する。再構成予測画像群蓄積部（２１０９）では、複数枚の再構成予測画像と各画像の撮影位置を蓄積し、出力画像選択部（２１１０）に出力する。参照画像復号化部（２１０５）では、符号化データを参照画像に復号化し、再構成参照画像群蓄積部（２１０８）に出力し、要求画像の撮影位置を出力画像選択部（２１１０）に出力する。予測画像復号化部（２１０６）では、符号化データを再構成参照画像を用いて予測画像に復号化し、再構成予測画像群蓄積部（２１０９）に出力し、要求画像の撮影位置を出力画像選択部（２１１０）に出力する。参照予測画像復号化部（２１０７）では、符号化データを参照画像と予測画像に復号化し、再構成した参照画像を再構成参照画像群蓄積部（２１０８）に出力し、再構成した予測画像を再構成予測画像群蓄積部（２１０９）に出力し、要求画像の撮影位置を出力画像選択部（２１１０）に出力する。出力画像選択部（２１１０）は、要求画像の撮影位置に対応する画像を再構成参照画像群蓄積部（２１０８）と再構成予測画像群蓄積部（２１０９）の中から取得し、画像表示部（２１１１）に出力する。画像表示部（２１１１）では、入力された画像を表示する。

【０１１５】以上のように構成された本実施の形態の画像復号化装置の動作を説明する。データ記憶・伝送部＊

＊（２１０１）では、ランダムアクセステーブルと符号化データを蓄積・伝送し、ランダムアクセステーブルを符号化データ選択部（２１０４）に出力する。また、符号化データ選択部（２１０４）により、符号化データが読み出される。

【０１１６】画像要求部（２１０２）は、マウスを持った計算機であり、ユーザがマウスを使って見たい画像を要求し、要求画像の撮影位置を画像蓄積検査手段（２１０３）に出力する。画像蓄積検査部（２１０３）では、要求された画像の撮影位置と、最大Ｎ枚蓄積できる再構成参照画像群蓄積部（２１０８）に蓄積された画像の撮影位置と最大Ｍ枚蓄積できる再構成予測画像群蓄積部（２１０９）に蓄積された画像の撮影位置により各画像に対する重要度を計算し、再構成参照画像群蓄積部（２１０８）に蓄積された画像のＮ番目に重要度が大きいものと再構成予測画像群蓄積部（２１０９）に蓄積された画像のＭ番目に重要度が大きいものを消去する。$\rho-\theta$空間上で距離の近いほど重要度は高いので、要求された画像の撮影位置をＰ、蓄積された画像の撮影位置をＱとすると、重要度はＷは、

【０１１７】

【数８】

$$W=\frac{1}{d(\mathbf{P},\mathbf{Q})}$$

【０１１８】で計算できる。ここで、

【０１１９】

【数９】

$$\mathbf{P}_1=(\rho_1,\theta_1),\ \mathbf{P}_2=(\rho_2,\theta_2)$$

【０１２０】とすると、

【０１２１】

【数１０】

$$d(\mathbf{P}_1,\mathbf{P}_2)=\min((\rho_1-\rho_2)^2,(\rho_1-\rho_2-360)^2,(\rho_1-\rho_2+360)^2)+(\theta_1-\theta_2)^2$$

【０１２２】である。符号化データ選択部（２１０４）では、図２２の状態テーブルに応じて図２３の出力データテーブルにしたがって、出力データを作成し、それぞれの出力先に出力する。図２２と図２３においてＰは要求画像の撮影位置であり、Ｑは要求画像が予測画像の場合に復号化に必要となる参照画像の撮影位置である。出力データは先頭に出力画像撮影位置データを配置し、続いて画像の再構成に必要なデータを配置している。

【０１２３】参照画像復号化部（２１０５）では、入力されたデータの先頭の出力画像撮影位置データを出力画像選択部（２１１０）に出力し、参照符号化データを参照画像に復号化し、再構成参照画像群蓄積部（２１０８）に出力する。

【０１２４】予測画像復号化部（２１０６）では、入力されたデータの先頭の出力画像撮影位置データを出力画像選択部（２１１０）に出力し、参照画像データを用いて予測符号化データを予測画像に復号化し、再構成予測画像群蓄積部（２１０９）に出力する。

【０１２５】参照予測画像復号化部（２１０７）では、入力されたデータの先頭の出力画像撮影位置データを出力画像選択部（２１１０）に出力し、参照符号化データを参照画像に復号化し、再構成された参照画像を用いて予測符号化データを予測画像に復号化し、再構成参照画像群蓄積部（２１０８）に再構成参照画像を出力し、再構成予測画像群蓄積部（２１０９）に再構成予測画像を出力する。

【０１２６】出力画像選択部（２１１０）では、入力された出力画像撮影位置データに対応する画像を再構成参照画像群蓄積部（２１０８）と再構成予測画像群蓄積部（２１０９）から取得し、画像表示部（２１１１）に出力する。

【０１２７】画像表示部（２１１１）では、入力された画像を表示する。以上のように、本実施の形態では、多視点画像復号化装置では、現在要求されている画像の撮

影位置の近傍の画像の要求頻度が高いことを利用し、要求された画像の撮影位置に近いものに高い重要度を与え、一度復号化した画像のうち重要度の高いものを蓄積しておくことで、効率的に画像を復号化することが可能である。

【０１２８】図２４は本発明の第８の実施の形態である画像復号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【０１２９】同図において、データ記憶・伝送部（２４０１）では、ランダムアクセステーブルと符号化データを蓄積・伝送し、ランダムアクセステーブルと符号化データを符号化データ選択部（２４０５）と予測符号化データ選択部（２４０９）に出力する。画像要求部（２４０２）は、要求する画像の撮影位置を画像要求蓄積部（２４０３）に出力する。画像要求蓄積部（２４０３）は、画像要求を蓄積し、符号化データ選択部（２４０５）により画像要求が読み出される。前回画像要求蓄積部（２４０４）は、前回処理した画像の撮影位置を蓄積し、符号化データ選択部（２４０５）によって、前回画像要求が読み出され、更新される。

【０１３０】符号化データ選択部（２４０５）は、前回処理した画像の要求や画像の蓄積の状態から復号化すべき符号化データを決定し、ランダムアクセステーブルと符号化データから必要な符号化データや画像データを選択し、参照予測画像復号化部１（２４０６）や参照画像復号化部１（２４０７）や予測画像復号化部１（２４０８）や画像表示部（２４１５）に出力する。

【０１３１】予測符号化データ選択部（２４０９）は、復号化されていない近傍の撮影位置を予測し、その位置の画像の復号化に必要なデータを、ランダムアクセステーブルと符号化データと画像データからを選択し、参照予測画像復号化部２（２４１０）や参照画像復号化部２（２４１１）や予測画像復号化部２（２４１２）に出力する。参照予測画像復号化部１（２４０６）は、符号化データから予測画像を復号化し、画像表示部（２４１５）に出力する。

【０１３２】参照画像復号化部１（２４０７）は、符号化データから参照画像を復号化し、画像表示部（２４１５）に出力する。予測画像復号化部１（２４０８）は、符号化データと画像データから予測画像を復号化し、画像表示部（２４１５）に出力する。参照予測画像復号化部２（２４１０）は、符号化データから参照画像と予測画像を復号化し、再構成参照画像群蓄積部（２４１３）と再構成予測画像群蓄積部（２４１４）に出力する。参照画像復号化部２（２４１１）は、符号化データから参照画像を復号化し、再構成参照画像群蓄積部（２４１３）に出力する。予測画像復号化部２（２４１２）は、符号化データと画像データから予測画像を復号化し、再構成予測画像群蓄積部（２４１４）に出力する。

【０１３３】以上のように構成された本実施の形態の画像復号化装置の動作を説明する。データ記憶・伝送部（２４０１）では、ランダムアクセステーブルと符号化データを蓄積・伝送し、ランダムアクセステーブルを符号化データ選択部（２４０５）に出力する。また、符号化データ選択部（２４０５）により、符号化データを読み出される。画像要求部（２４０２）は、マウスを持った計算機であり、ユーザがマウスを使って見たい画像を要求し、要求画像の撮影位置を画像要求蓄積手段（２４０３）に出力する。画像要求蓄積部（２４０３）では、要求された画像の撮影位置を蓄積し、符号化データ選択部（２４０５）によって、撮影位置を読み出される。

【０１３４】前回要求画像蓄積部（２４０４）では、処理した撮影位置を蓄積し、符号化データ選択部（２４０５）によって、撮影位置を読み出される。符号化データ選択部（２４０５）では、図２５の状態テーブルに応じて図２６の出力データテーブルにしたがって、出力データを作成し、それぞれの出力先へ出力する。図２５と図２６においてＰは要求画像の撮影位置であり、Ｑは要求画像が予測画像の場合に復号化に必要となる参照画像の撮影位置であり、Ｒは前回処理された画像の撮影位置である。さらに、Ｒ≠Ｐならば、再構成予測画像群蓄積部（２４１３）と再構成参照画像群蓄積部（２４１４）に蓄積されている画像を消去し、Ｐを前回要求画像蓄積部（２４０４）に書き込む。

【０１３５】参照予測画像復号化部１（２４０６）は、参照符号化データと予測符号化データを予測画像に復号化し、画像表示部（２４１５）に出力する。参照画像復号化部１（２４０７）は、参照符号化データを参照画像に復号化し、画像表示部（２４１５）に出力する。予測画像復号化部１（２４０８）は、参照画像データと予測符号化データを予測画像に復号化し、画像表示部（２４１５）に出力する。

【０１３６】予測符号化データ選択部（２４０９）では、図２７の予測テーブルの上から順に、再構成参照画像群蓄積部（２４１３）と再構成予測画像群蓄積部（２４１４）に蓄積されていないか検索し、蓄積されていないものがあれば、その撮影位置を予測撮影位置としてＳとして、図２８の状態テーブルに応じて図２９の出力データテーブルにしたがって、出力データを作成しそれぞれの出力先へ出力する。図２８と図２９で、Ｔは、撮影位置Ｓの画像が予測画像の場合の復号化に必要な参照画像の撮影位置である。

【０１３７】参照予測画像復号化部２（２４１０）は、参照符号化データと予測符号化データを参照画像と予測画像に復号化し、参照画像は再構成参照画像群蓄積部（２４１３）に、予測画像は再構成予測画像群蓄積部（２４１４）に出力する。参照画像復号化部２（２４１１）は、参照符号化データを参照画像に復号化し、再構成参照画像群蓄積部（２４１３）に出力する。

【０１３８】予測画像復号化部２（２４１２）は、参照

画像データと予測符号化データを予測画像に復号化し、再構成予測画像群蓄積部（２４１４）に出力する。画像表示部（２４１５）は、入力された画像を表示する。

【０１３９】以上のように、本実施の形態では、多視点画像復号化装置では、現在要求されている画像の撮影位置の近傍の画像の要求頻度が高いことを利用し、画像の要求の変化がないときには、要求された画像の撮影位置に近いもの復号化しておくことで、効率的に画像を復号化することが可能である。

【０１４０】図３０は本発明の第９の実施の形態である画像符号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【０１４１】同図において、撮影位置制御型画像撮像部（３００１）は、対象物体の画像を撮影し、その撮影位置データと画像データを出力する。参照予測グループ部（３００５）は、撮影位置データと画像データを参照画像と一枚以上の予測画像からなる参照予測グループにグループ化し、参照画像を参照画像符号化部（３００６）に出力し、撮影位置データと参照画像と予測画像を変形拘束式推定部（３００８）に出力する。

【０１４２】参照画像符号化部（３００６）は、参照画像を符号化し、再構成参照画像を作成し、符号化データを参照関係作成部（３０１０）に出力し、再構成参照画像を再構成参照画像蓄積部（３００７）に出力する。再構成参照画像蓄積部（３００７）は、再構成参照画像を蓄積し、拘束予測画像符号化部（３００９）に再構成参照画像を出力する。拘束式推定部（３００８）は、拘束式を推定し、拘束予測画像符号化部（３００９）に出力する。

【０１４３】拘束予測画像符号化部（３００９）は、予測画像を符号化し、符号化データを参照関係作成部（３０１０）に出力する。参照関係テーブル作成部（３０１０）は、符号化データから参照関係と符号化データ位置と撮影位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データをデータ記録・伝送部（３０１１）に出力する。データ記録・伝送部（３０１１）は、用途に応じて記録・伝送する。

【０１４４】以上のように構成された本実施の形態の画像符号化装置の動作を説明する。まず、画像位置制御型撮像部（３００１）では、図１の画像位置制御型撮像部と同様に、対象物体の周囲を周囲を網羅するように、異なる視点から画像を撮影し、撮影位置・画像データを取得し、図３に示す書式で出力する。

【０１４５】参照グループ化部（３００５）では、図１の参照グループ化部（１０５）と同様に、参照画像とその近傍の画像を予測画像としてグループ化し、インデックス順に図５に示す書式で、参照画像符号化部（３００６）と変形拘束式推定部（３００８）へ出力する。参照画像符号化部（３００６）は、図１の参照画像符号化部（１０６）と同様である。再構成参照画像蓄積部（３００７）は、図１の再構成参照画像蓄積部（１０７）と同様である。

【０１４６】変形拘束式推定部（３００８）では、エピポーラ拘束と呼ばれる拘束式のパラメータ７つを推定する（「エピポラー　ジェオメトリ　イン　ステレオ　モーション　アンド　オブジェクト　レコグニション」"Epipolar Geometry in Stereo、 Motion and Object Recognition - A Unified Approach、" Gang Xu and Zheng you Zhang、 Kluwer Academic Publishers、 1996））。

【０１４７】図３１はエピポーラ拘束の説明図であり、３次元空間中の点Ｍを異なる視点から撮影した場合である。Ｃ１、Ｃ２はそれぞれの画像１と画像２を撮影したカメラの焦点を示し、Ｍは、画像１上ではｍ１に、画像２上ではｍ２に投影される。Ｃ２は画像１上のｅ１に、Ｃ１は画像２上のｅ２に投影され、ｅ１とｅ２はそれぞれの画像上のエピ極と呼ばれる。３次元空間中のＣ１とｍ１を通る直線を画像２上への投影した直線をエピポーラ線と呼び、エピポーラ線２として図３１に示す。

【０１４８】エピポーラ線２は、画像１を撮影したカメラの内部パラメータ（焦点距離、光軸）と視点位置、画像２を撮影したカメラの内部パラメータと視点位置、画像１上の点の座標により決定できる。エピポーラ線２を決定できれば、画像１上の点ｍ１に３次元空間で対応する画像２上の点ｍ２の位置は、Ｃ１からＭの距離が得られない場合でも、エピポーラ線２上に拘束できる。

【０１４９】図３２に示す二つの視点から撮影された画像１と画像２において、画像１上の点（ｘ１、ｙ１）に対応する画像２上のエピポーラ線は以下のように表わされる。

【０１５０】

【数１１】

$$(x_1, y_1, 1)\mathbf{F}(x, y, 1)^{\mathbf{T}} = 0$$

【０１５１】ここで、

【０１５２】

【数１２】

$$\mathbf{F} = \mathbf{A}^{-T}\mathbf{t}_{\times}\mathbf{R}\mathbf{A}^{-1}$$

【０１５３】

【数１３】

$$\mathbf{A} = \begin{pmatrix} f & 0 & ox \\ 0 & f & oy \\ 0 & 0 & 1 \end{pmatrix}$$

【０１５４】

【数１４】

25 26

$$R=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos(\rho_2-\rho_1) & -\sin(\rho_2-\rho_1)\\ 0 & \sin(\rho_2-\rho_1) & \cos(\rho_2-\rho_1)\end{pmatrix}\begin{pmatrix}\cos\theta_2 & 0 & -\sin\theta_2\\ 0 & 1 & 0\\ \sin\theta_2 & 0 & \cos\theta_2\end{pmatrix}\begin{pmatrix}\cos\theta_1 & 0 & \sin\theta_1\\ 0 & 1 & 0\\ -\sin\theta_1 & 0 & \cos\theta_1\end{pmatrix}$$

【０１５５】

【数１５】

$$\mathbf{t}=\mathbf{R}(0,0,-d)^T+(0,0,d)^T$$

【０１５６】であり、ｏｘ、ｏｙはカメラの光軸の画像上の座標を表わし、ｆは焦点距離を示す。

【０１５７】（$\rho 1$，$\theta 1$）、（$\rho 2$，$\theta 2$）、ｄは、図３２に示す撮影位置を示し、×は、

【０１５８】

【数１６】

$$\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}_\times=\begin{pmatrix}0 & -x_3 & x_2\\ x_3 & 0 & -x_1\\ -x_2 & x_1 & 0\end{pmatrix}$$

【０１５９】の演算を示す。カメラの内部パラメータであるｏｘ、ｏｙ、ｆは、予め求めておき、（$\rho 1$，$\theta 1$）、（$\rho 2$，$\theta 2$）、ｄは撮影システムにより与えられる。７つのパラメータｅは、

【０１６０】

【数１７】

$$\mathbf{F}=\begin{pmatrix}f_{11} & f_{12} & f_{13}\\ f_{21} & f_{22} & f_{23}\\ f_{31} & f_{32} & f_{33}\end{pmatrix}$$

【０１６１】とすると、

【０１６２】

【数１８】

$$\mathbf{e}=\begin{pmatrix}\frac{f_{32}f_{21}-f_{22}f_{31}}{f_{22}f_{11}-f_{21}f_{12}}\\ \frac{f_{31}f_{12}-f_{11}f_{32}}{f_{22}f_{11}-f_{21}f_{12}}\\ \frac{f_{23}f_{12}-f_{22}f_{13}}{f_{22}f_{11}-f_{21}f_{12}}\\ \frac{f_{13}f_{21}-f_{11}f_{23}}{f_{22}f_{11}-f_{21}f_{12}}\\ \frac{-f_{12}}{f_{21}}\\ \frac{f_{11}}{f_{21}}\\ \frac{-f_{22}}{f_{21}}\end{pmatrix}$$

【０１６３】で計算できる。予測画像の撮影位置を（$\rho 1$，$\theta 1$）とし、その予測画像を予測するための参照画像（同じインデックスのついた参照画像）の撮影位置（$\rho 2$，$\theta 2$）として用いて、それぞれの参照予測の組に対してエピポーラパラメータを推定する。入力である図５のデータの予測画像にエピポーラパラメータを付加し、図３３に示す書式で、拘束予測画像符号化部（３００９）へ出力する。

【０１６４】拘束予測画像符号化部（３００９）のフローチャートを図３４に示す。ブロック化ステップでは、予測画像データをブロックに分割する。

【０１６５】エピポーラ線計算ステップでは、各ブロックの中心点の座標を（ｘ１，ｙ１）とし、拘束式推定部（３００８）で推定されたエピポーラパラメータｅからＦを以下のように計算し、式１１を用いてブロックにエピポーラ線を求める。

【０１６６】

【数１９】

$$\mathbf{F}=\begin{pmatrix}b & a & -(ay_b+bx_b)\\ -1 & -c & (cy_b+bx_b)\\ y_a-bx_a & cy_a-ax_a & -(cy_b-x_b)y_a+(ay_b+bx_b)x_a\end{pmatrix}$$

【０１６７】ここで、

【０１６８】

【数２０】

27

$$e = \begin{pmatrix} x_a \\ y_a \\ x_b \\ y_b \\ a \\ b \\ c \end{pmatrix}$$

【０１６９】である。動き推定ステップでは、図３５に示すように、予測画像のブロックをエピポーラ線に沿って、再構成参照画像上を１次元的に移動させて、予測画像ブロックと再構成参照画像の画素値の差の絶対値の総和が最小になる１次元の移動量を求める。

【０１７０】ブロック残差計算ステップでは、予測画像ブロックと再構成参照画像の画素値の差の絶対値の総和が最小になる位置での、画素値の差を計算する。ＤＣＴステップでは、ブロック残差計算ステップで計算されたブロック残差をデジタルコサイン変換する。量子化ステップでは、ＤＣＴステップの出力結果を所定の量子幅で量子化する。

【０１７１】可変長符号化ステップでは、動き推定ステップで計算された１次元移動量と、量子化ステップの出力結果を可変長符号化し、符号化データとして図３６に示す書式で出力する。

【０１７２】参照関係テーブル作成部（３０１０）は、参照関係テーブル作成部（１０９）と同様である。データ記録・伝送部（３０１１）は、データ記録・伝送部（１１０）と同様である。

【０１７３】以上のように、本実施の形態では、多視点画像のように、動きのないシーンを与えられた視点から撮影した場合、２つの画像の対応点が撮影対象によらず、カメラの視点と内部パラメータから決定できるある直線上に拘束されることを利用して、符号化データに拘束パラメータを付加する一方で、２次元の動き情報を一次元の動き情報に減少させることで、効率の良い符号化が可能である。

【０１７４】図３７は本発明の第１０の実施の形態である画像復号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【０１７５】同図において、データ記憶・伝送部（３７０１）では、参照関係テーブルと符号化データを蓄積・伝送し、参照テーブルと符号化データを符号化データ選択部（３７０２）に出力する。符号化データ選択部（３７０２）では、参照関係テーブルからデータの位置を検索し、符号化データから必要なデータを参照画像復号化部（３７０３）や拘束予測画像復号化部（３７０５）に出力する。参照画像復号化部（１７０３）では、参照画像符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（３７０４）に出力し、必要なら画像表示手段（３７０６）にも出力する。

28

【０１７６】再構成参照画像蓄積部（３７０５）は、再構成参照画像を蓄積し、拘束予測画像復号化部（３７０５）に出力する。拘束予測画像復号化部（３７０５）では、予測画像符号化データを復号化し、再構成予測画像を画像表示部（３７０６）に出力する。画像表示部（３７０６）は入力された画像を蓄積しておき、要求により表示する。

【０１７７】以上のように構成された本実施の形態の画像復号化装置の動作を説明する。データ記憶・伝送部（３７０１）では、参照関係テーブルと符号化データを蓄積・伝送し、参照テーブルと符号化データを符号化データ選択部（３７０２）に出力する。符号化データ選択部（３７０２）では、参照関係テーブルから符号化データ位置を検索し、データ記憶・伝送部（３７０１）から、符号化データ位置の符号化データを取得する。参照／予測フラグと参照符号化データを、参照画像復号化部（３７０３）に出力し、予測符号化データがあれば、予測画像復号化部（３７０５）に出力する。

【０１７８】ただし、参照画像復号化部（３７０３）に出力する参照符号化データが前回と同じ場合は、参照画像はすでに復号化され、蓄積されているので、参照画像復号化部（３７０３）への出力は省略する。

【０１７９】参照画像復号化部（３７０３）では、参照画像が符号化された参照符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（３７０４）に出力する。また、参照／予測フラグにおいて、参照画像であれば、画像表示部（３７０６）に再構成参照画像を出力する。

【０１８０】再構成参照画像蓄積部（３７０４）では、再構成された参照画像を蓄積し、拘束予測画像復号化部（３７０５）に出力する。拘束予測画像復号化部（３７０５）のフローチャートを図３８に示す。復号化ステップでは、可変長符号化されたエピポーラパラメータ、動き情報、残差情報を復号化する。

【０１８１】エピポーラ線計算ステップでは、ブロックの中心座標とエピポーラパラメータから、図３４のエピポーラ線計算ステップと同様にエピポーラ線を計算する。動き補償ステップでは、エピポーラ線に沿って、復号化された移動量だけ移動したところからブロックを得、変形画像を構成する。逆量子化ステップでは、残差情報を逆量子化する。逆ＤＣＴステップでは、逆量子化された残差情報を逆ＤＣＴし、画素の残差情報を得る。ブロック再構成ステップでは、逆ＤＣＴステップで得られた残差情報を、動き補償ステップで得られた変形画像に加えて予測画像データを再構成する。

【０１８２】以上のように予測符号化データを再構成参照画像を用いて復号化し、画像表示部（３７０６）に再構成予測画像を出力する。画像表示部（３７０６）では、復号化された画像を蓄積しておき、計算機に接続されたマウスにより検索し、表示する。

29

【０１８３】以上のように、本実施の形態では、多視点画像のように、動きのないシーンを与えられた視点から撮影した場合、２つの画像の対応点が撮影対象によらず、カメラの視点と内部パラメータから決定できるある直線上に拘束されることを利用して、符号化データに拘束パラメータを付加する一方で、２次元の動き情報を一次元の動き情報に減少させることで、効率の良く符号化した符号化データの復号化が可能である。

【０１８４】図３９は本発明の第１１の実施の形態である画像符号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【０１８５】同図において、撮影位置光源位置制御型画像撮像部（３９０１）は、対象物体の画像を撮影し、その撮影位置データと光源位置データと画像データを出力する。参照予測グループ部（３９０７）は、撮影位置データと光源位置データと画像データを参照画像と一枚以上の予測画像からなる参照予測グループにグループ化し、参照画像を参照画像符号化部（３９０８）に出力し、予測画像を予測画像符号化部（３９０８）に出力する。

【０１８６】参照画像符号化部（３９０８）は、参照画像を符号化し、再構成参照画像を作成し、符号化データを参照関係作成部（３９１１）に出力し、再構成参照画像を再構成参照画像蓄積部（３９０９）に出力する。再構成参照画像蓄積部（３９０９）は、再構成参照画像を蓄積し、予測画像符号化部（３９１０）に再構成参照画像を出力する。

【０１８７】予測画像符号化部（３９１０）は、予測画像を符号化し、符号化データを参照関係作成部（３９１１）に出力する。参照関係テーブル作成部（３９１１）は、符号化データから参照関係と符号化データ位置と撮影位置と光源位置を記述した参照関係テーブルを作成し、参照関係テーブルと符号化データをデータ記録・伝送部（３９１０）に出力する。データ記録・伝送部（３９１２）は、用途に応じて記録・伝送する。

【０１８８】以上のように構成された本実施の形態の画像符号化装置の動作を説明する。撮影位置光源位置制御型画像撮像部（３９０１）では、光源位置が対象物体の周囲を網羅するように、異なる光源位置で画像を撮影する。計算機（３９０４）によって、サーボモータ２（３９０４）を対象物体の正面から水平方向にρ１度、水平面から垂直方向にθ１度の位置に光源（３９０５）を移動させる。さらに、それぞれの光源位置について、カメラ位置が対象物体の周囲を網羅するように、異なる視点から画像を撮影する。

【０１８９】計算機（３９０４）によってサーボモータ１（３９０２）を対象物体の正面から水平方向にρｃ度、水平面から垂直方向にθｃ度の位置にカメラ（３９０３）を移動させ、撮影位置・光源位置・画像データを計算機（３９０４）によって取得する。ここで、得られる画像データは図２と同様に輝度プレーン、色差１プレーン、色差２プレーン、アルファプレーンで構成する。ここで、ρ１，θ１，ρｃ，θｃは、

30

【０１９０】

【数２１】

$$0 \leq \rho_c < 360,\ 0 \leq \theta_c \leq 90,\ (\rho_c, \theta_c \text{は整数})$$

【０１９１】

【数２２】

$$0 \leq \rho_l < 360,\ 0 \leq \theta_l \leq 90,\ (\rho_l, \theta_l \text{は10の倍数})$$

【０１９２】とする。このように取得された撮影位置・光源位置・画像データは、図４０に示すように、各画像データの先頭に水平方向撮影角度ρｃと、垂直方向撮影角度θｃと、水平方向撮影角度ρ１と、垂直方向撮影角度θ１を付加した書式で出力される。

【０１９３】参照予測グループ化部（３９０７）では、参照画像によって類似した画像を予測画像として予測符号化するために、ρ１－θ１－ρｃ－θｃ空間において、等間隔に参照画像を配置し、参照画像の近傍の画像をその参照画像より予測される予測画像にグループ化する。参照予測グループ化部（３９０７）での処理は、図１の参照予測グループ化部（１０５）の処理を４次元的に行う。グループ化された撮影位置・光源位置・画像データはインデックス順に図４１の示す書式で、参照画像は参照画像符号化部（３９０８）へ、予測画像は予測画像符号化部（３９１０）へ出力する。

【０１９４】参照画像符号化部（３９０８）では、参照画像を図１の参照画像符号化部（１０６）と同様に符号化し、符号化データを参照関係テーブル作成部（３９１１）に出力し、参照画像の再構成画像を作成し、再構成参照画像蓄積部（３９０９）に出力する。

【０１９５】再構成参照画像蓄積部（３９０９）では、再構成参照画像を蓄積しておき、予測画像符号化部（３９１０）に出力する。参照予測グループ化部（３９０７）において、図４１のように参照画像と予測画像が整列されて出力されるので、再構成参照画像蓄積部（３９０９）は、画像１枚分の容量を持っていれば十分である。

【０１９６】予測画像符号化部（３９１０）では、図１の予測画像符号化部（１０８）と同様に、予測画像を再構成参照画像を用いて符号化し、符号化データを参照関係テーブル作成部（３９１１）に出力する。参照関係テーブル作成部（３９１１）では、各符号化部より出力された図４２のようなデータを順に走査し、図４３のような参照関係テーブルと図４４のような符号化データを作成し、データ記録・伝送部（３９１２）に出力する。

【０１９７】データ記録・伝送部（３９１２）では、参照関係テーブルと符号化データを磁気ディスクや光ディスクに蓄積しておいたり、ネットワークなどで伝送する。

【０１９８】以上のように、本実施の形態では、異なる

多くの視点から異なる多くの光源位置で撮影された多視点画像を、撮影位置データが近傍の類似した画像である参照画像と１枚以上の予測画像にグループ化し、効率よく符号化することが可能である。さらに、画像を各参照予測グループで参照画像が先頭になるように整列して、符号化手段に出力することで、計算量と計算に必要な蓄積容量を少なくしている。また、その整列順に符号化データ位置を格納したテーブルを付加することで、ランダムアクセスに対応した効率的な復号化が可能である。

【０１９９】図４５は本発明の第１２の実施の形態である画像復号化装置の構成を示すブロック図であり、同図を用いて本実施の形態の構成を説明する。

【０２００】同図において、データ記憶・伝送部（４５０１）では、参照関係テーブルと符号化データを蓄積・伝送し、参照テーブルと符号化データを符号化データ選択部（４５０２）に出力する。符号化データ選択部（４５０２）では、参照関係テーブルから要求された画像のデータの位置を検索し、符号化データから必要なデータを参照画像復号化部（４５０３）や予測画像復号化部（４５０５）に出力する。

【０２０１】要求された光源位置の符号化データがなければ、近傍にある複数の光源位置の符号化データを選択し、それぞれを参照画像復号化部（４５０３）や予測画像復号化部（４５０５）に出力する。参照画像復号化部（４５０３）では、参照画像符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（４５０４）と中間画像生成部（４５０６）に出力する。再構成参照画像蓄積部（４５０５）は、再構成参照画像を蓄積し、予測画像復号化部（４５０５）に出力する。予測画像復号化部（４５０５）では、予測画像符号化データを復号化し、再構成予測画像を中間画像生成部（４５０６）に出力する。

【０２０２】中間画像生成部（４５０６）では、要求された光源位置の画像がなければ、近傍の光源位置の画像を用いて要求された光源位置の画像の画像を生成し、画像表示部（４５０７）に出力する。画像表示部（４５０７）は入力された画像を蓄積しておき、要求により表示する。

【０２０３】以上のように構成された本実施の形態の画像復号化装置の動作を説明する。データ記憶・伝送部（４５０１）では、参照関係テーブルと符号化データを蓄積・伝送し、参照テーブルと符号化データを符号化データ選択部（４５０２）に出力する。

【０２０４】符号化データ選択部（４５０２）では、要求された光源位置と撮影位置を参照関係テーブルから符号化データ位置を検索し、データ記憶・伝送部（４５０１）から、符号化データ位置の符号化データを取得する。参照／予測フラグと参照符号化データを、参照画像復号化部（４５０３）に出力し、予測符号化データがあれば、予測画像復号化部（４５０５）に出力する。

【０２０５】ただし、参照画像復号化部（４５０３）に出力する参照符号化データが前回と同じ場合は、参照画像はすでに復号化され、蓄積されているので、参照画像復号化部（４５０３）への出力は省略する。また、要求された光源位置のデータがない場合には、光源位置が近傍の４つの符号化データをそれぞれ参照画像復号化部（４５０３）と予測画像復号化部（４５０４）へ出力する。

【０２０６】要求された画像が、（$\rho 1$，$\theta 1$，ρc，θc）であり、参照関係テーブルに、光源位置が水平・垂直方向の角度が１０の倍数の画像しか無い場合、近傍の４つの符号化データ（$\rho 11$，$\theta 11$，ρc，θc）、（$\rho 11$，$\theta 12$，ρc，θc）、（$\rho 12$，$\theta 11$，ρc，θc）、（$\rho 12$，$\theta 12$，ρc，θc）は、以下の式によって得られる。

【０２０７】

【数２３】

$$\rho_{l1} = \lceil \rho_l \rfloor$$

【０２０８】

【数２４】

$$\rho_{l2} = \lceil \rho_l \rfloor + 10$$

【０２０９】

【数２５】

$$\theta_{l1} = \lceil \theta_l \rfloor$$

【０２１０】

【数２６】

$$\theta_{l2} = \lceil \theta_l \rfloor + 10$$

【０２１１】ここで、「・」は、・を越えない１０の倍数を示す。また、$\rho 12$が３６０になる場合は、$\rho 12 = 0$とする。

【０２１２】参照画像復号化部（４５０３）では、参照画像が符号化された参照符号化データを復号化し、再構成参照画像を再構成参照画像蓄積部（４５０４）に出力する。

【０２１３】また、参照／予測フラグにおいて、参照画像であれば、中間画像生成部（４５０６）に再構成参照画像を出力する。再構成参照画像蓄積部（４５０４）では、再構成された参照画像を蓄積し、予測画像復号化部（４５０５）に出力する。

【０２１４】予測画像復号化部（４５０５）では、予測画像が符号化された予測符号化データを再構成参照画像を用いて復号化し、中間画像生成部（４５０６）に再構成予測画像を出力する。中間画像生成部（４５０６）では、要求された光源位置と復号化された画像データの光源位置が等しければ、画像表示部（４５０６）へ出力する。等しくなければ、符号化データ選択部（４５０２）で選択された光源位置が近傍の画像より中間画像を生成する。

【０２１５】要求された画像の光源位置を（$\rho 1$，$\theta 1$，ρc，θc）とし、符号化データ選択部（４５０

33

２）で選択された画像の光源位置を（ρｌ１，θｌ１，ρｃ，θｃ）、（ρｌ１，θｌ２，ρｃ，θｃ）、（ρｌ２，θｌ１，ρｃ，θｃ）、（ρｌ２，θｌ２，ρｃ，θｃ）とする。それぞれの近傍画像をＩ１、Ｉ２、Ｉ３、Ｉ４とし、画像Ｉ１の座標（ｘ，ｙ）の画素値を＊Ｉ１（ｘ，ｙ）と表わすと、中間画像Ｉの画素値Ｉ（ｘ，ｙ）を以下の式で計算する。

34

【０２１６】

【数２７】

$$I(x,y) = r_3(r_1 I_1(x,y) + r_2 I_3(x,y) + r_4(r_1 I_2(x,y) + r_2 I_4(x,y)$$

【０２１７】ここで、

【０２１８】

【数２８】

$$r_1 = \frac{\rho_{l2} - \rho_l}{\rho_{l2} - \rho_{l1}}$$

【０２１９】

【数２９】

$$r_2 = \frac{\rho_l - \rho_{l1}}{\rho_{l2} - \rho_{l1}}$$

【０２２０】

【数３０】

$$r_3 = \frac{\theta_{l2} - \theta_l}{\theta_{l2} - \theta_{l1}}$$

【０２２１】

【数３１】

$$r_4 = \frac{\theta_l - \theta_{l1}}{\theta_{l2} - \theta_{l1}}$$

【０２２２】である。生成された中間画像は画像表示部（４５０７）へ出力される。画像表示部（４５０７）では、復号化された画像を蓄積し、表示する。

【０２２３】以上のように、本実施の形態では、多くの異なる視点から多くの異なる光源位置で撮影した画像を、撮影位置が近傍の画像をグループ化し、グループを予測符号化された符号化データを、その参照予測関係と符号化データ位置と撮影位置を記述したテーブルを用いることにより、効率的に復号化することが可能である。

【０２２４】また、参照関係テーブルはグループ順に記述されているので、同一参照予測グループ内が連続的に復号化されるので、再構成参照画像を蓄積しておき、予測符号化データを復号化するだけで、効率的に画像を復号化することが可能である。

【０２２５】さらに、要求された光源位置の画像が符号化されていない場合でも、光源位置が近傍の画像から要求された光源位置の画像を生成するので、要求画像を表示することが可能である。

【０２２６】

【発明の効果】本発明の画像符号化装置と復号化装置では、以下の理由により、従来の画像符号化技術を用いるよりも効率の良い符号化と復号化が可能となる。

【０２２７】１．画像符号化データに参照関係と撮影位置と符号化データ位置を記述したテーブルを付加することにより、参照画像から複数の予測画像に対して予測符号化を用いて符号化する。

【０２２８】２．画像符号化データに参照関係と撮影位置と符号化データ位置を記述したテーブルを参照することにより、要求する画像の符号化データ位置を直ちに検索でき、復号化できる。

【０２２９】３．画像再生の要求を監視しておき、それを参照することにより、高速な要求の変化や計算能力に応じて、自動的に、画質は悪いが計算負荷の小さい変形画像と画質は良いが計算負荷の大きい予測画像を段階的に復号化する。

【０２３０】４．蓄積すべき画像の撮影位置を計算し、復号化処理済みの一枚数の画像を蓄積しておき、復号化処理の回数を減らす。

【０２３１】５．再生すべき画像に変化がないときに、再生すべき画像の撮影位置と撮影位置が復号化する画像撮影位置を計算し、あらかじめ復号化処理しておき、再生すべき画像の変化に高速な応答をする。

【０２３２】６．撮影シーンに動きがない場合に、撮影対象によらず対応点が直線上に拘束できることを利用して、拘束パラメータと１次元的な動き情報で、画像を効率に符号化する。

【０２３３】７．撮影シーンに動きがない場合に、撮影対象によらず対応点が直線上に拘束できることを利用して、拘束パラメータと１次元的な動き情報で、画像を効率に符号化した符号化データを復号化する。

【０２３４】８．画像符号化データに参照関係と撮影位置と光源位置と符号化データ位置を記述したテーブルを付加することにより、多くの視点で多くの照明効果を持つ対象物体画像を、要求にたいして効率的に復号化可能な状態で符号化する。

【０２３５】９．画像符号化データに参照関係と撮影位置と光源位置と符号化データ位置を記述したテーブルが付加された、符号化データを、要求にたいして効率的に復号化する。さらに、要求される光源位置の画像が符号化データに無い場合でも、光源位置が近傍の画像から中間画像を生成する。

【図面の簡単な説明】

【図１】本発明の第１の実施の形態における画像符号化装置のブロック図

【図２】画像データの構成の図

【図３】撮影位置データと画像データの図

【図４】予測符号化の説明図

【図５】参照予測グループごとに整列された撮影位置データと画像データの図

【図６】参照画像符号化部のフローチャート

【図７】予測画像符号化部のフローチャート

35

【図８】参照予測グループごとに整列された撮影位置データと符号化データの図
【図９】参照関係テーブルの図
【図１０】符号化データの図
【図１１】第２の実施の形態における画像復号化装置のブロック図
【図１２】第３の実施の形態における画像復号化装置のブロック図
【図１３】ランダムアクセステーブルの図
【図１４】ランダムアクセスヘッダの構成の図
【図１５】第４の実施の形態における画像符号化装置のブロック図
【図１６】第５の実施の形態における画像復号化装置のブロック図
【図１７】第６の実施の形態における画像復号化装置のブロック図
【図１８】復号化状況蓄積テーブルの説明図
【図１９】符号化データ選択部（１７０５）の状態テーブルの図
【図２０】符号化データ選択部（１７０５）の出力データテーブルの図
【図２１】第７の実施の形態における画像復号化装置のブロック図
【図２２】符号化データ選択部（２１０４）の状態テーブルの図
【図２３】符号化データ選択部（２１０４）の出力データテーブルの図
【図２４】第８の実施の形態における画像復号化装置のブロック図
【図２５】符号化データ選択部（２４０５）の状態テーブルの図
【図２６】符号化データ選択部（２４０５）の出力データテーブルの図
【図２７】予測テーブルの図
【図２８】予測符号化データ選択部（２４０９）の状態テーブルの図
【図２９】予測符号化データ選択部（２４０９）の出力データテーブルの図
【図３０】第８の実施の形態における画像符号化装置のブロック図
【図３１】エピポーラ拘束の説明図
【図３２】異なる二つの撮影位置の説明図
【図３３】エピポーラパラメータの付加された画像データの図
【図３４】拘束予測符号化部のフローチャート
【図３５】拘束された動き推定方法の説明図
【図３６】拘束予測符号化データの図
【図３７】第９の実施の形態における画像復号化装置のブロック図
【図３８】拘束予測復号化部のフローチャート

36

【図３９】第１０の実施の形態における画像符号化装置のブロック図
【図４０】撮影位置と光源位置を付加した画像データの図
【図４１】参照予測グループごとに整列された撮影位置データと光源位置データと画像データの図
【図４２】参照予測グループごとに整列された撮影位置データと光源位置データと符号化データの図
【図４３】参照関係テーブル作成部（３９１１）の出力する参照関係テーブルの図
【図４４】参照関係テーブル作成部（３９１１）の出力する符号化データの図
【図４５】第１０の実施の形態における画像復号化装置のブロック図

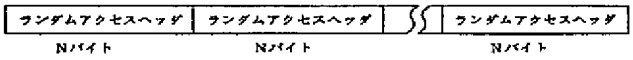
【符号の説明】

１０１　撮影位置制御型画像撮像部
１０２　サーボモータ
１０３　カメラ
１０４　計算機
１０５　参照予測グループ化部
１０６　参照画像符号化部
１０７　再構成参照画像蓄積部
１０８　予測画像符号化部
１０９　参照関係テーブル作成部
１１０　データ記録・伝送部
２０１　輝度プレーン
２０２　色差１プレーン
２０３　色差２プレーン
２０４　アルファプレーン
４０１　参照画像
４０２　予測画像
１１０１　データ記録・伝送部
１１０２　符号化データ選択部
１１０３　参照画像復号化部
１１０４　再構成参照画像蓄積部
１１０５　予測画像復号化部
１１０６　画像表示部
１２０１　データ記録・伝送部
１２０２　ランダムアクセステーブル作成部
１２０３　符号化データ選択部
１２０４　画像要求部
１２０５　参照画像復号化部
１２０６　再構成参照画像蓄積部
１２０７　予測画像復号化部
１２０８　画像表示部
１５０１　撮影位置制御型画像撮像部
１５０２　参照予測グループ化部
１５０３　参照画像符号化部
１５０４　再構成参照画像蓄積部
１５０５　予測画像符号化部

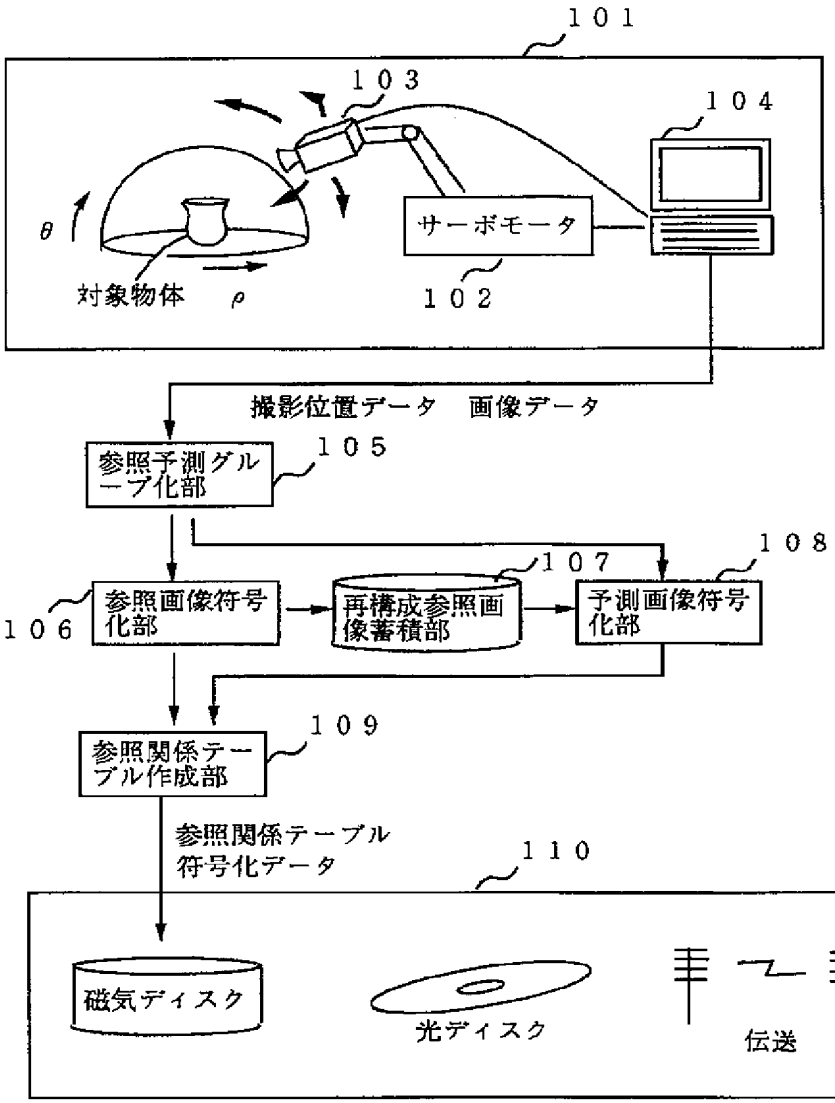
１５０６　参照関係テーブル作成部
１５０７　ランダムアクセステーブル作成部
１５０８　データ記録・伝送部
１６０１　データ記録・伝送部
１６０２　符号化データ選択部
１６０３　画像要求部
１６０４　参照画像復号化部
１６０５　再構成参照画像蓄積部
１６０６　予測画像復号化部
１６０７　画像表示部
１７０１　データ記録・伝送部
１７０２　画像要求監視部
１７０３　画像要求蓄積部
１７０４　復号化状況蓄積部
１７０５　符号化データ選択部
１７０６　参照画像復号化部
１７０７　再構成参照画像蓄積部
１７０８　変形参照画像復号化部
１７０９　変形参照画像蓄積部
１７１０　予測画像復号化部
１７１１　再構成予測画像蓄積部
１７１２　出力画像選択部
１７１３　画像表示部
１８０１　復号化状況テーブル
１８０２　復号化レベルテーブル
２００１　出力データ書式
２００２　出力データテーブル
２１０１　データ記録・伝送部
２１０２　画像要求部
２１０３　画像蓄積検査部
２１０４　符号化データ選択部
２１０５　参照画像復号化部
２１０６　予測画像復号化部
２１０７　参照予測画像復号化部
２１０８　再構成参照画像群蓄積部
２１０９　再構成予測画像群蓄積部
２１１０　出力画像選択部
２１１１　画像表示部
２４０１　データ記録・伝送部
２４０２　画像要求監視部
２４０３　画像要求蓄積部
２４０４　前回画像要求蓄積部
２４０５　符号化データ選択部
２４０６　参照画像復号化部１
２４０７　予測画像復号化部１

２４０８　参照予測画像復号化部１
２４０９　予測符号化データ選択部
２４１０　参照画像復号化部２
２４１１　予測画像復号化部２
２４１２　参照予測画像復号化部２
２４１３　再構成参照画像群蓄積部
２４１４　再構成予測画像群蓄積部
２４１５　画像表示部
３００１　撮影位置制御型画像撮像部
３００２　サーボモータ
３００３　カメラ
３００４　計算機
３００５　参照予測グループ化部
３００６　参照画像符号化部
３００７　再構成参照画像蓄積部
３００８　変形拘束式推定部
３００９　拘束予測画像符号化部
３０１０　参照関係テーブル作成部
３０１１　データ記録・伝送部
３７０１　データ記録・伝送部
３７０２　符号化データ選択部
３７０３　参照画像復号化部
３７０４　再構成参照画像蓄積部
３７０５　拘束予測画像復号化部
３７０６　画像表示部
３９０１　撮影位置光源位置制御型画像撮像部
３９０２　サーボモータ１
３９０３　カメラ
３９０４　サーボモータ２
３９０５　光源
３９０６　計算機
３９０７　参照予測グループ化部
３９０８　参照画像符号化部
３９０９　再構成参照画像蓄積部
３９１０　予測画像符号化部
３９１１　参照関係テーブル作成部
３９１２　データ記録・伝送部
４５０１　データ記録・伝送部
４５０２　符号化データ選択部
４５０３　参照画像復号化部
４５０４　再構成参照画像蓄積部
４５０５　予測画像復号化部
４５０６　中間画像生成部
４５０７　画像表示部

【図１３】



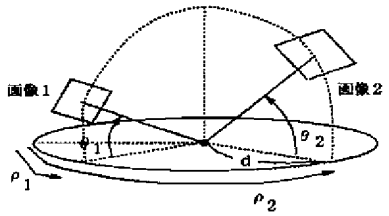
【図１】



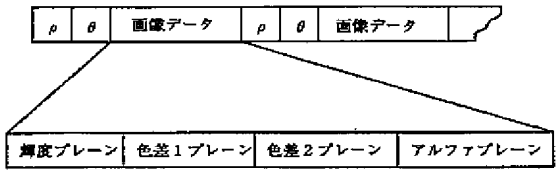
【図２７】

P+ (0, 1)
P+ (1, 0)
P+ (0, −1)
P+ (−1, 0)
P+ (1, 1)
P+ (1, −1)
P+ (−1, −1)
P+ (−1, 1)

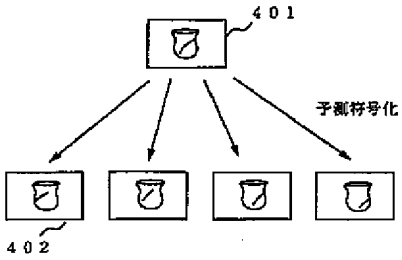
【図３２】



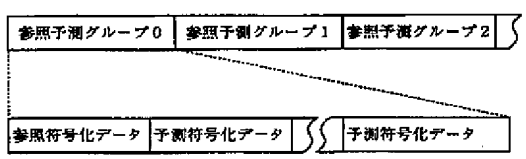
【図３】



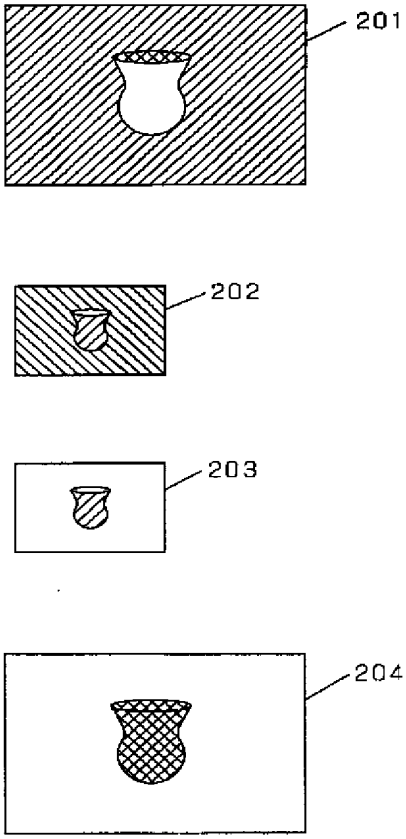
【図４】



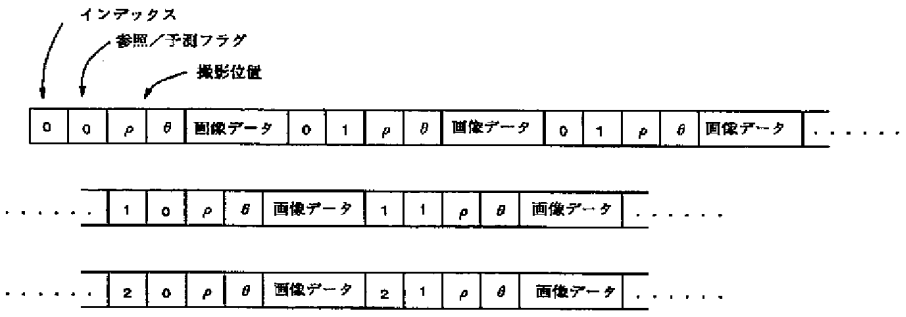
【図１０】



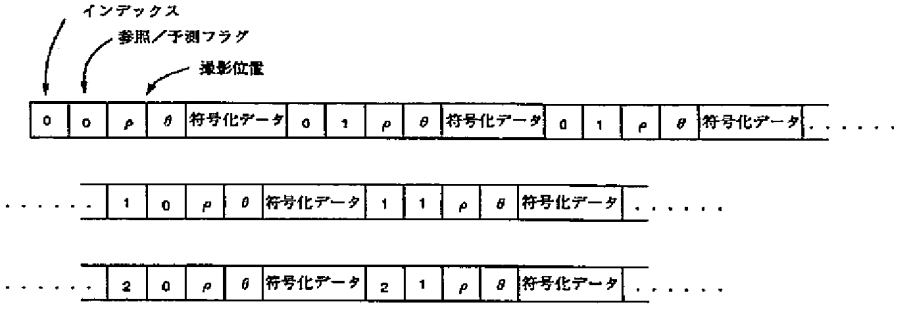
【図２】



【図５】



【図８】

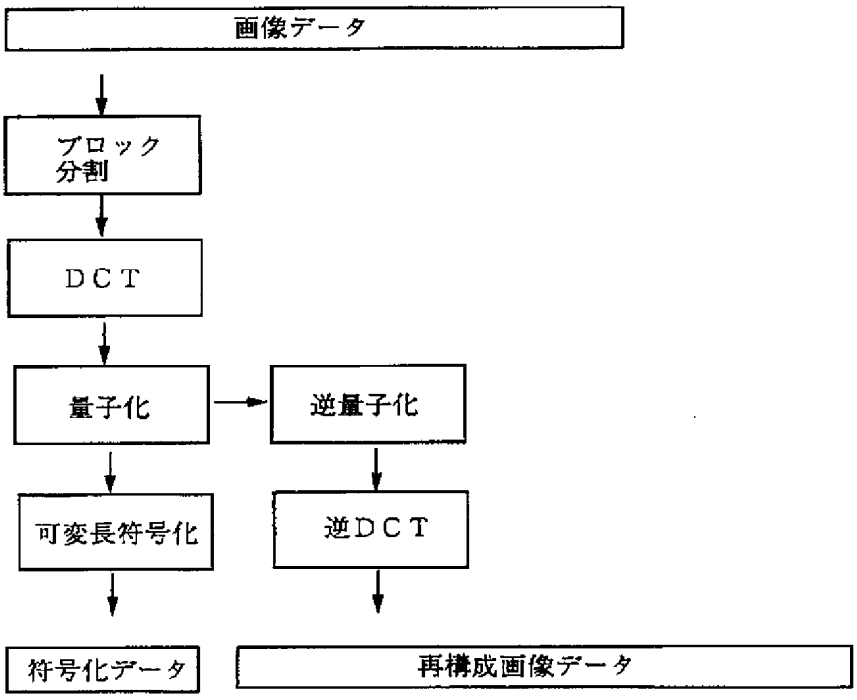


【図１８】

参照画像撮影位置	予測画像撮影位置	復号化レベル

1801

【図６】



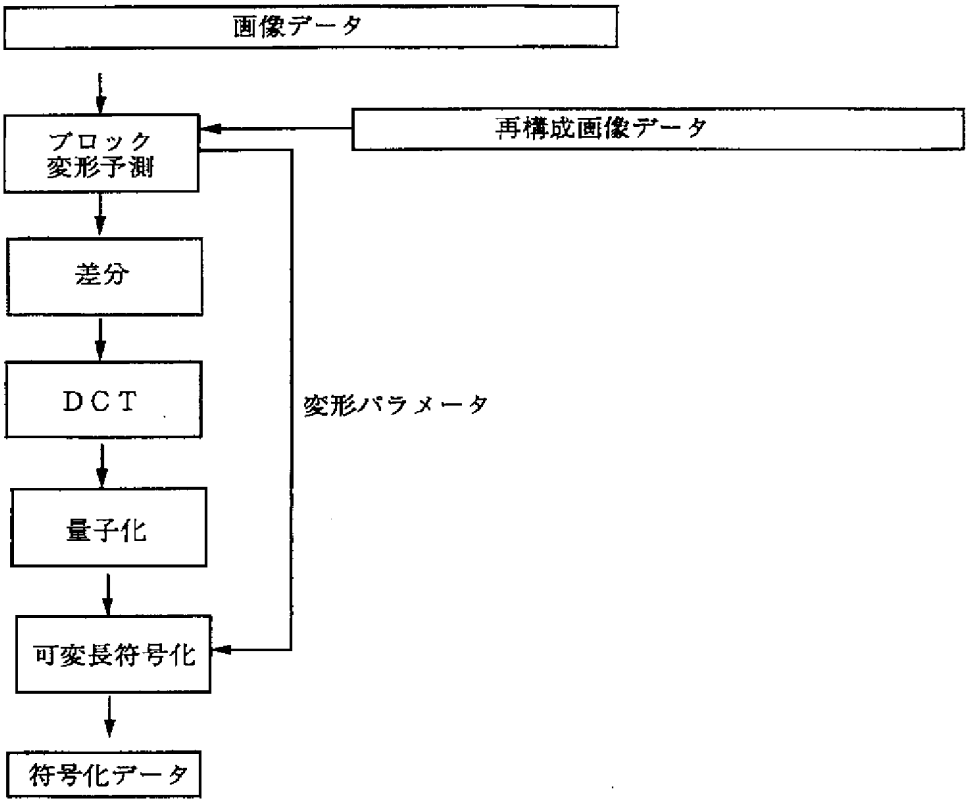
1802

復号化レベル	復号化状況
1	変形参照画像復号化済み
2	予測画像復号化済み

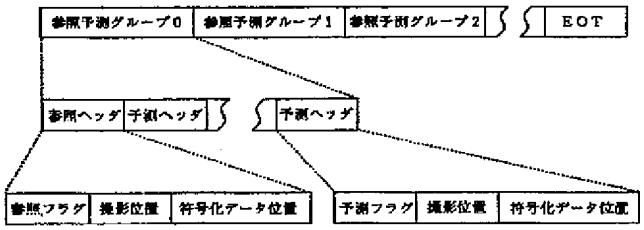
【図１９】

1	位置Ｐは参照画像	Ｐ＝Ｒ		
2		Ｐ≠Ｒ		
3	位置Ｐは予測画像	Ｑ≠Ｒ		
4		Ｑ＝Ｒ	Ｐ≠Ｓ	
5			Ｐ＝Ｓ	Ｌ＝１
6				Ｌ＝２

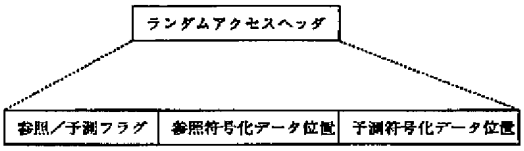
【図７】



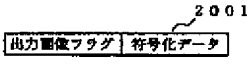
【図９】



【図１４】



【図２０】



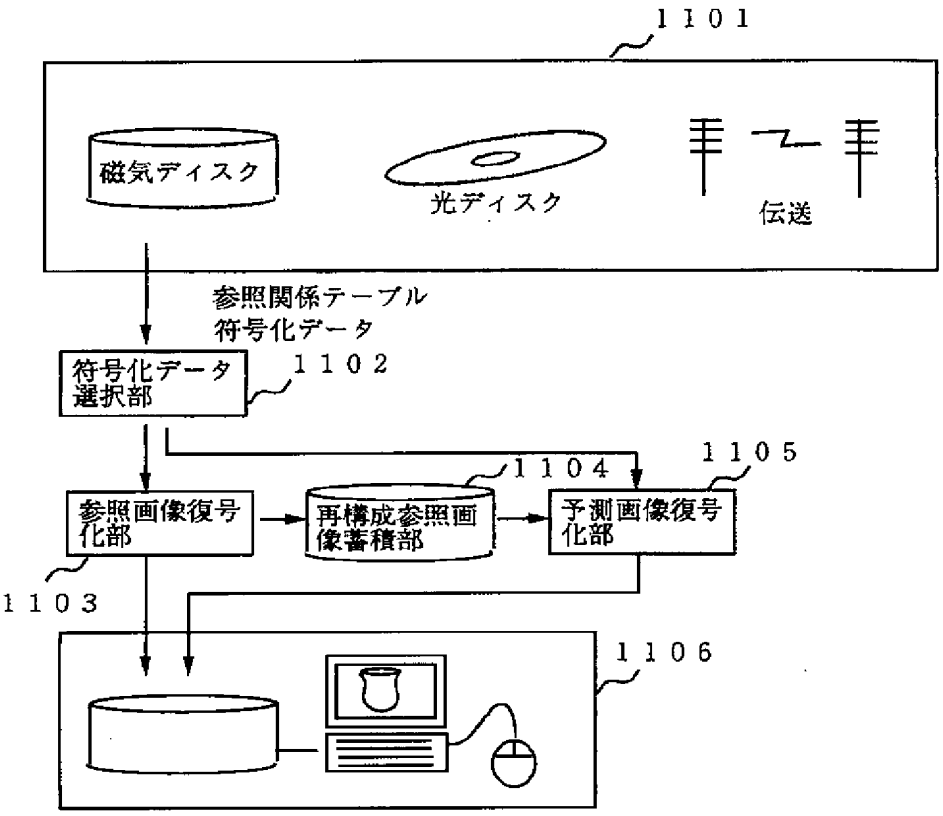
【図２２】

1	位置Ｐは参照画像	Ｐは蓄積されている	
2		Ｐは蓄積されていない	
3	位置Ｐは予測画像	Ｐは蓄積されていない	Ｑは蓄積されている
4			Ｑは蓄積されていない
5		Ｐは蓄積されている	

2002

状態	出力データ	出力先	復号化状況テーブル
1	1	出力画像選択部	R S L
2	1 参照符号化データ（P）	参照画像復号化部	P S L
3	0 参照符号化データ（Q）	参照画像復号化部	Q S L
4	2 予測符号化データ（P）	変形画像復号化部	R P 1
5	3 予測符号化データ（P）	予測画像復号化部	R P 2
6	3	出力画像選択部	R S L

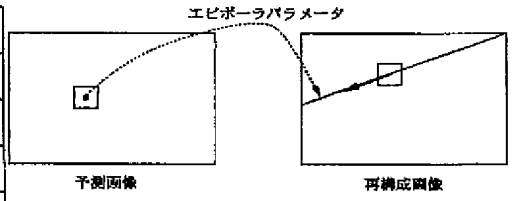
【図１１】



【図２３】

状態	出力先	出力データ
1	出力画像選択部	P
2	参照画像復号化部	P | 参照符号化データ（P）
3	予測画像復号化部	P | 参照画像データ（Q） | 予測符号化データ（P）
4	参照予測画像復号化部	P | 参照符号化データ（Q） | 予測符号化データ（P）
5	出力画像選択部	P

【図３５】



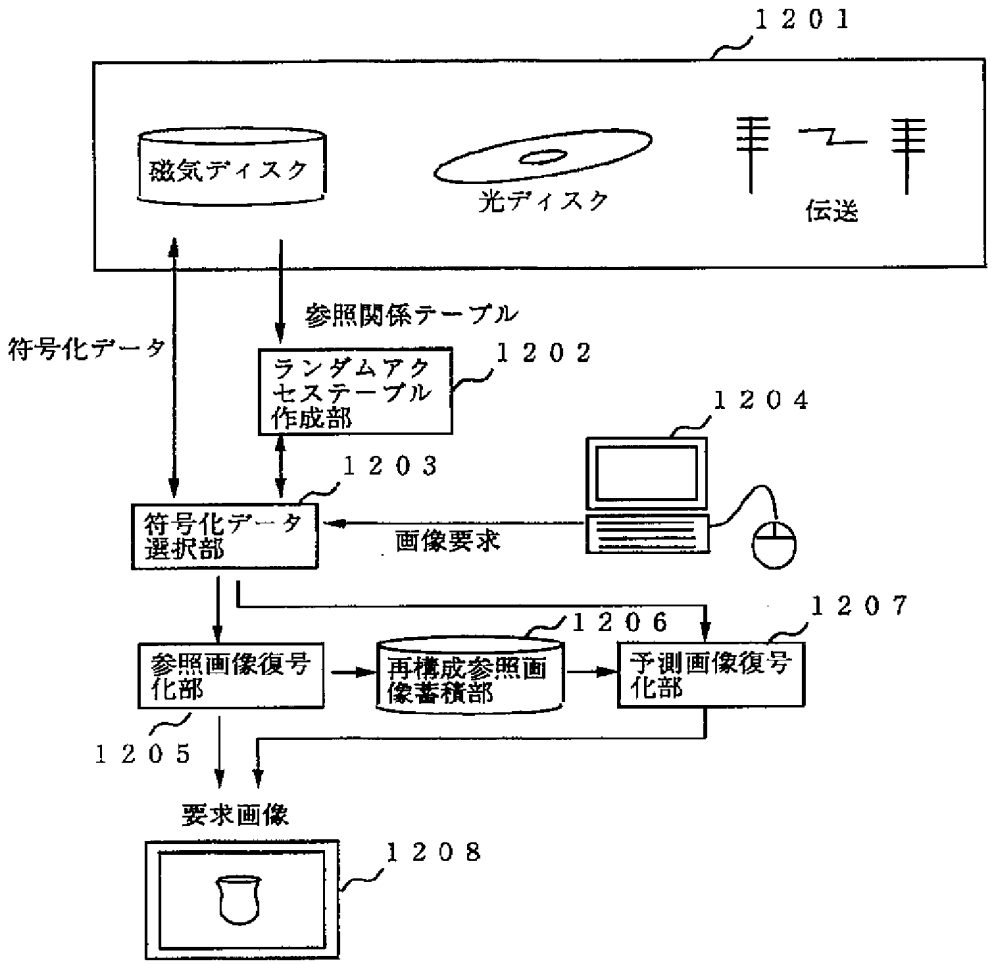
【図２５】

1	P＝R			
2	P≠R	位置Pは参照画像	Pは蓄積されている	
3			Pは蓄積されていない	
4		位置Pは予測画像	Pは蓄積されていない	Qは蓄積されている
5				Qは蓄積されていない
6			Pは蓄積されている	

【図２８】

1	位置Sは参照画像	Sは蓄積されていない	
2	位置Sは予測画像	Sは蓄積されていない	Tは蓄積されている
3			Tは蓄積されていない

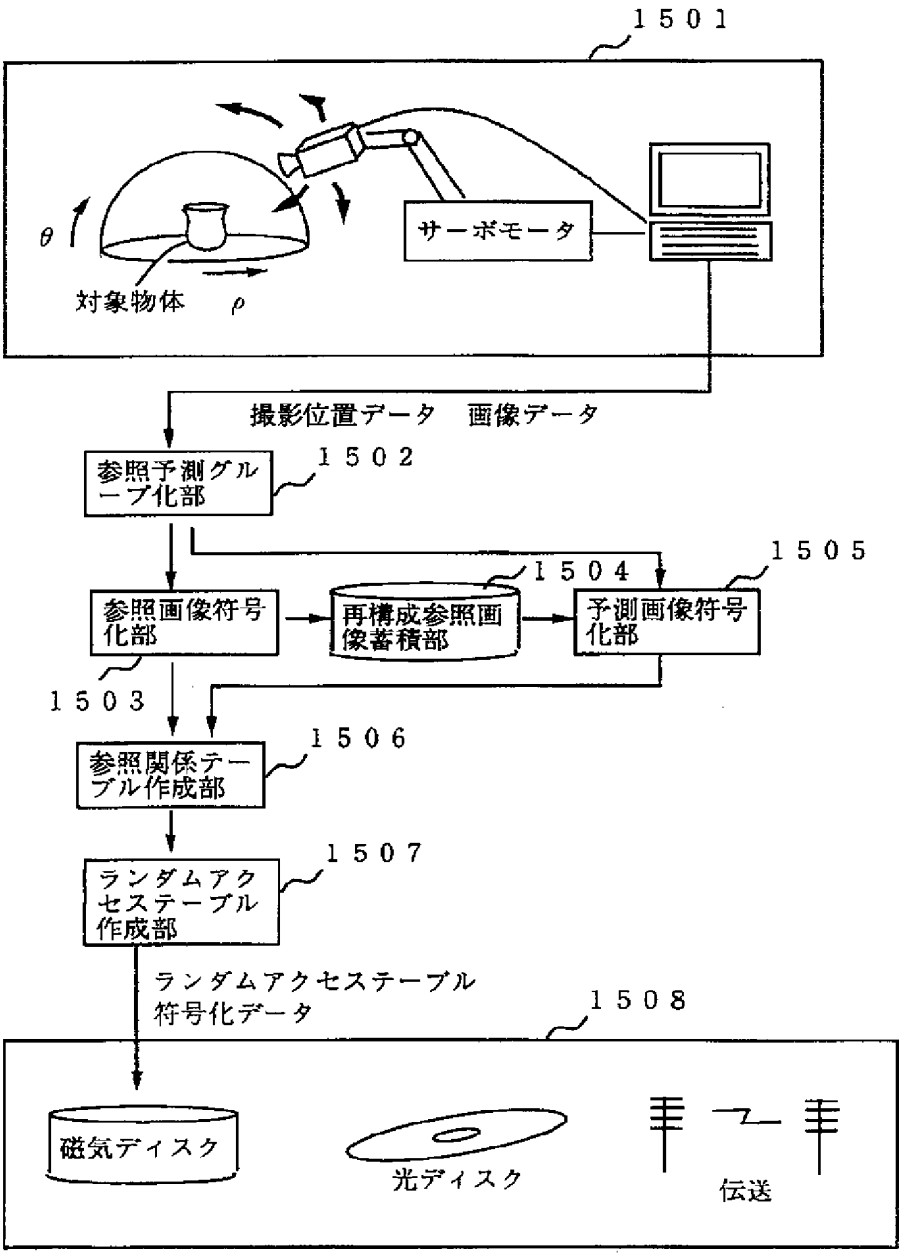
【図１２】



【図２６】

決順	出力先	出力データ
1	予測符号化データ選択部	なし
2	画像表示部	画像データ（P）
3	参照画像復号化部	参照符号化データ（P）
4	予測画像復号化部	参照画像データ（Q）　予測符号化データ（P）
5	参照予測画像復号化部	参照符号化データ（Q）　予測符号化データ（P）
6	画像表示部	画像データ（P）

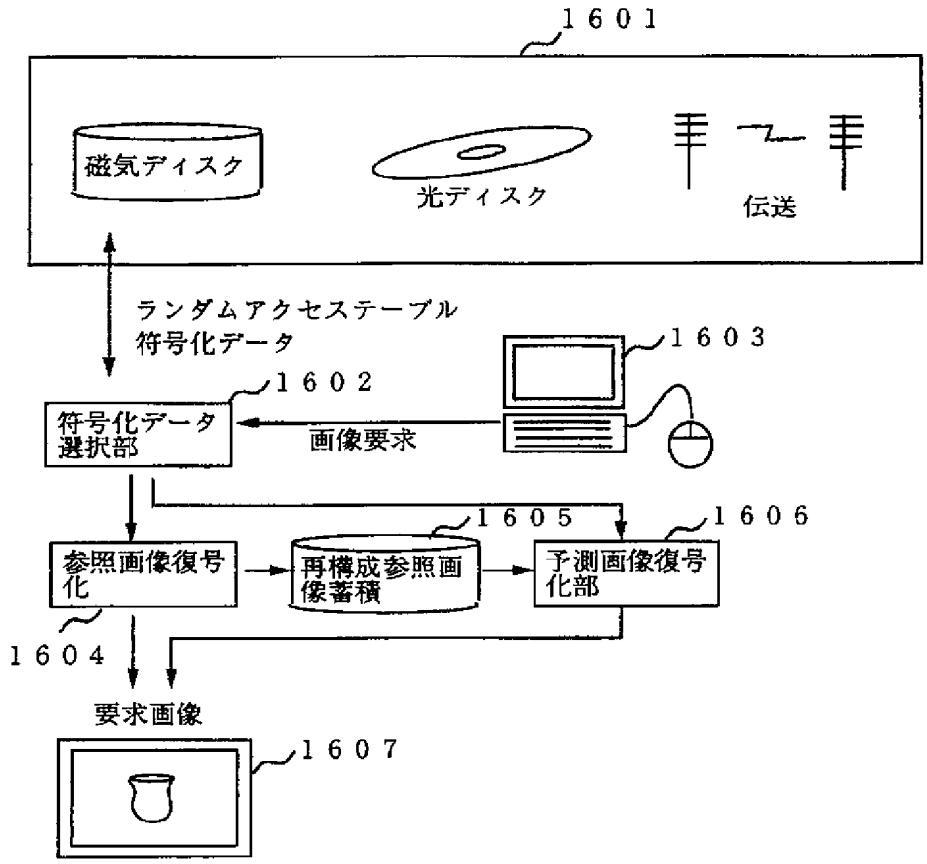
【図１５】



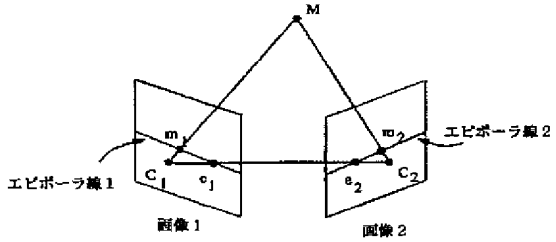
【図２９】

状態	出力先	出力データ
1	参照画像復号化部	参照符号化データ（P）
2	予測画像復号化部	参照画像データ（Q）　予測符号化データ（P）
3	参照予測画像復号化部	参照符号化データ（Q）　予測符号化データ（P）

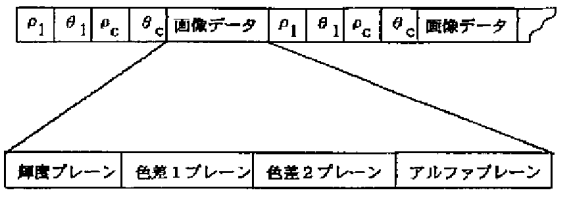
【図１６】



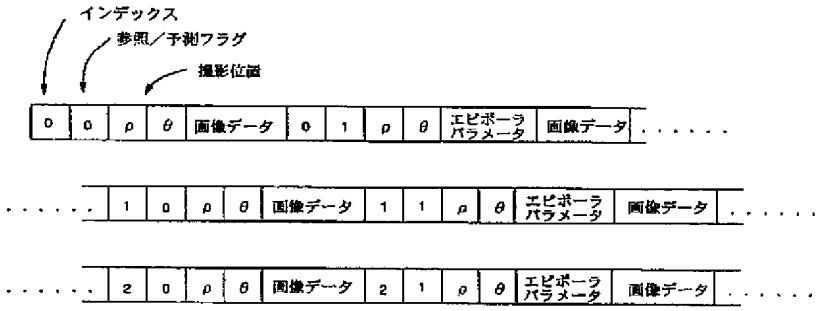
【図３１】



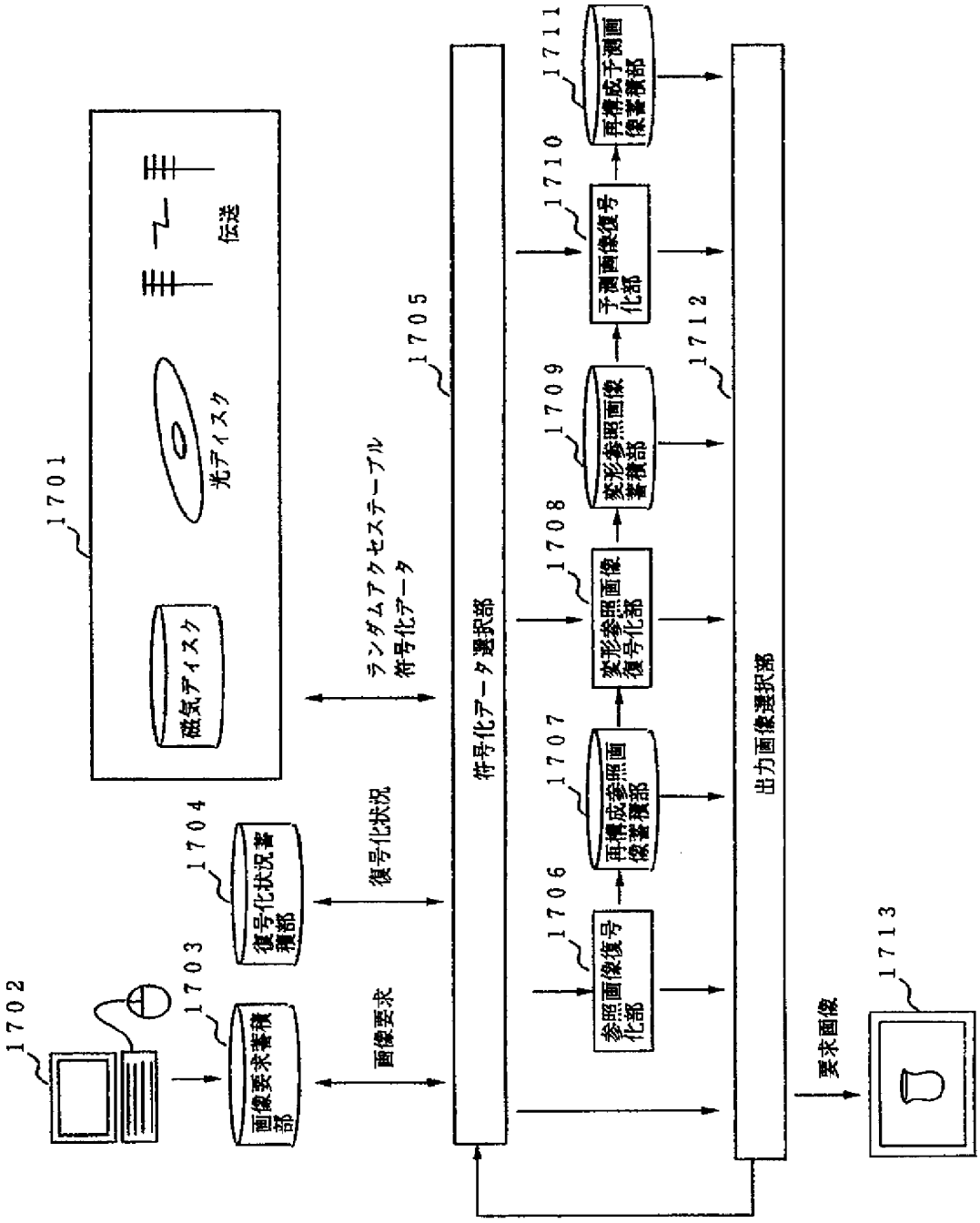
【図４０】



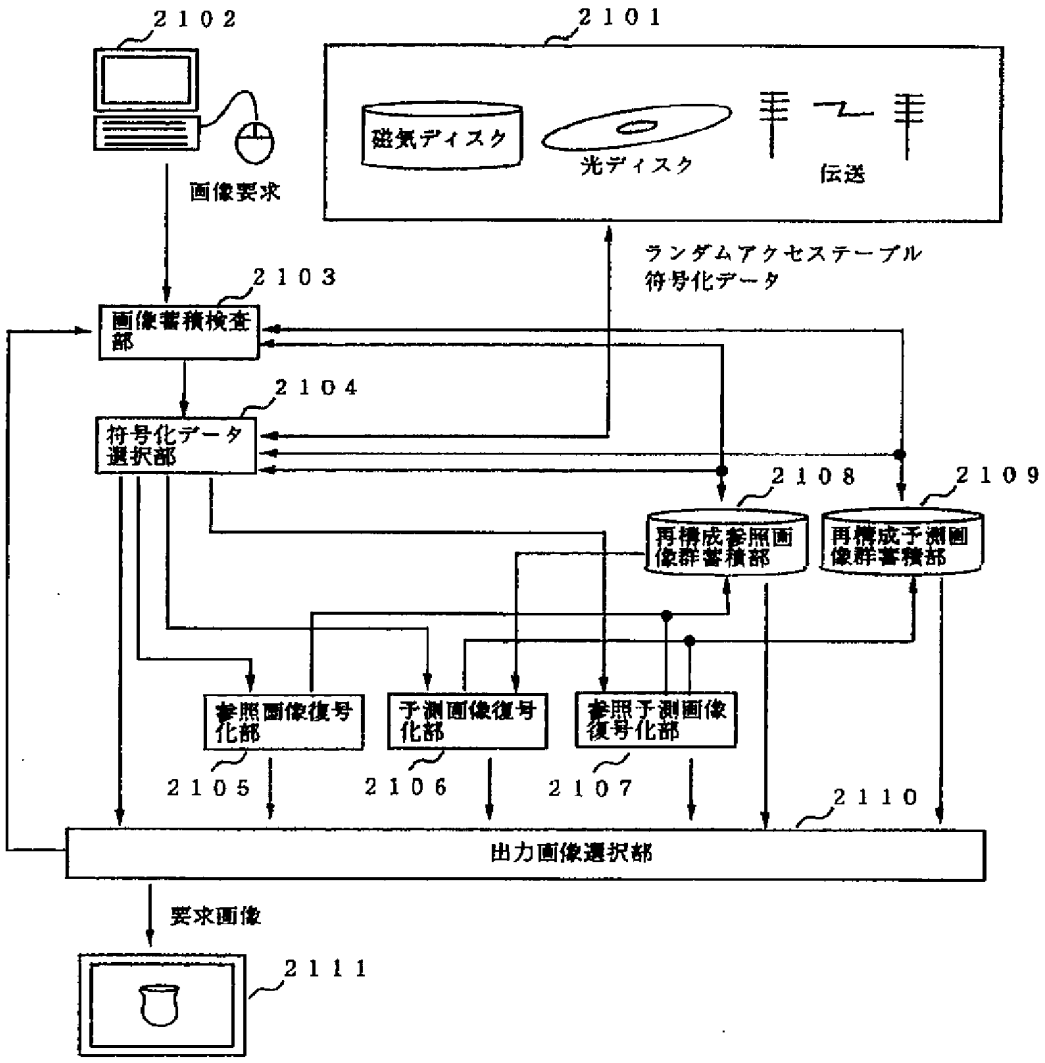
【図３３】



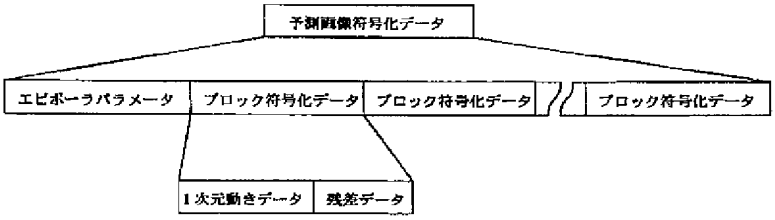
【図１７】


1701
磁気ディスク
光ディスク
伝送
ランダムアクセステーブル
符号化データ
1705
符号化データ選択部
1702
1703
画像要求蓄積部
画像要求
1704
復号化状況蓄積部
復号化状況
1706
参照画像復号化部
1707
圧構成参照画像蓄積部
1708
変形参照画像復号化部
1709
変形参照画像蓄積部
1710
予測画像復号化部
1711
画像成予測画像蓄積部
1712
出力画像選択部
1713
要求画像

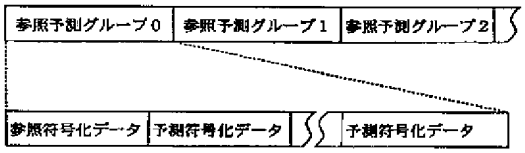
【図２１】


２１０２
２１０１
磁気ディスク
光ディスク
伝送
画像要求
ランダムアクセステーブル
符号化データ
２１０３
画像蓄積検査部
２１０４
符号化データ選択部
２１０８
２１０９
再構成参照画像群蓄積部
再構成予測画像群蓄積部
参照画像復号化部
予測画像復号化部
参照予測画像復号化部
２１０５
２１０６
２１０７
２１１０
出力画像選択部
要求画像
２１１１

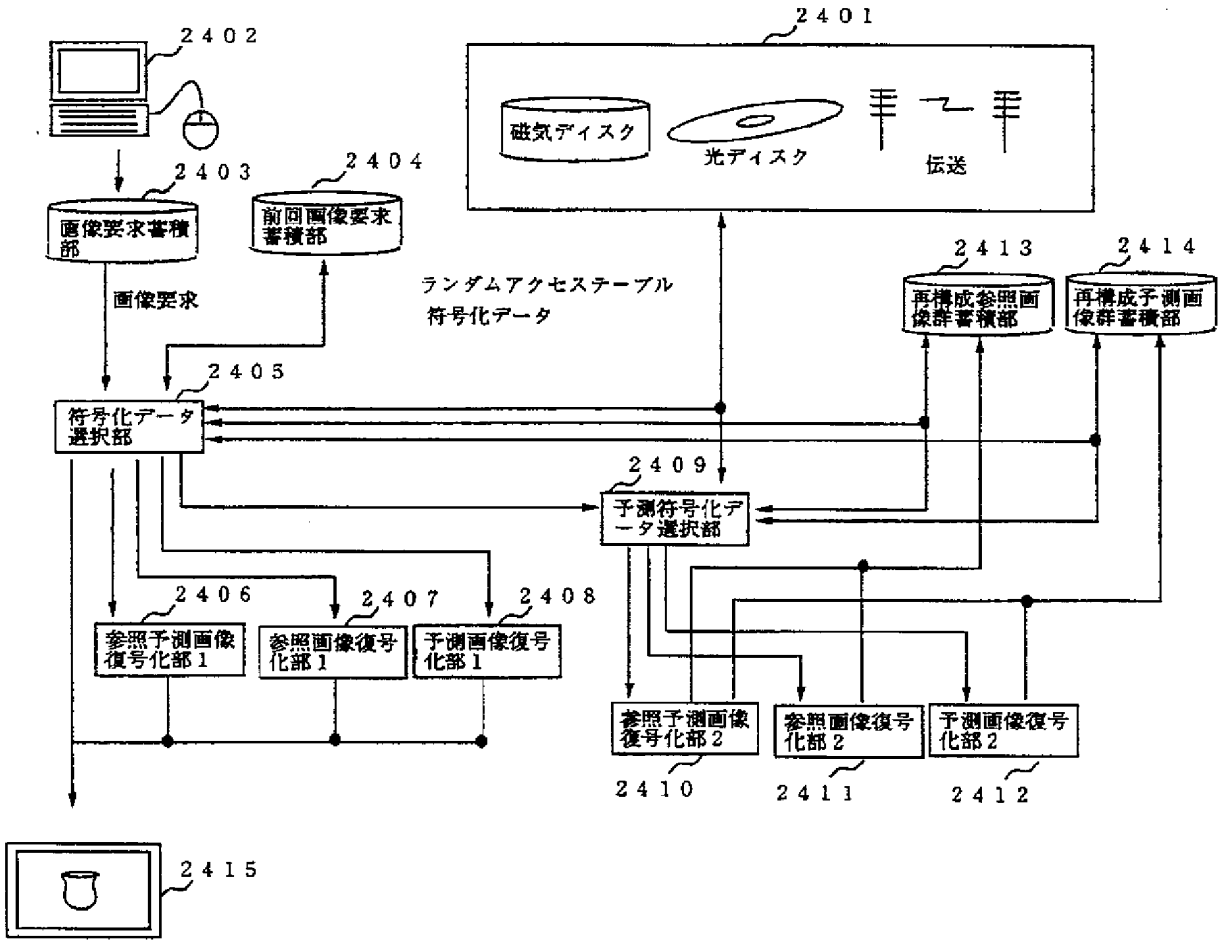
【図３６】


予測画像符号化データ
エピポーラパラメータ
ブロック符号化データ
ブロック符号化データ
ブロック符号化データ
1次元動きデータ
残差データ

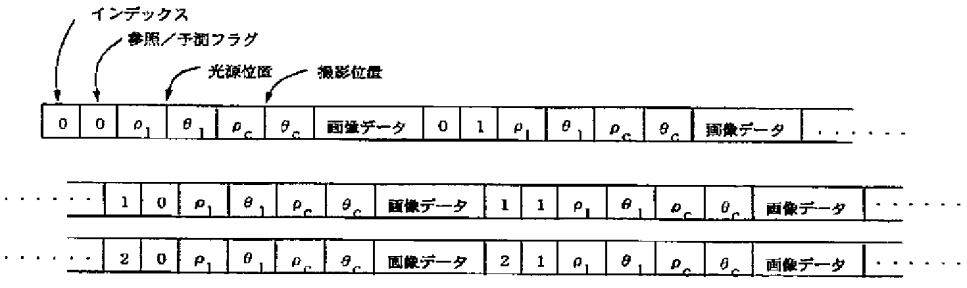
【図４４】


参照予測グループ０
参照予測グループ１
参照予測グループ２
参照符号化データ
予測符号化データ
予測符号化データ

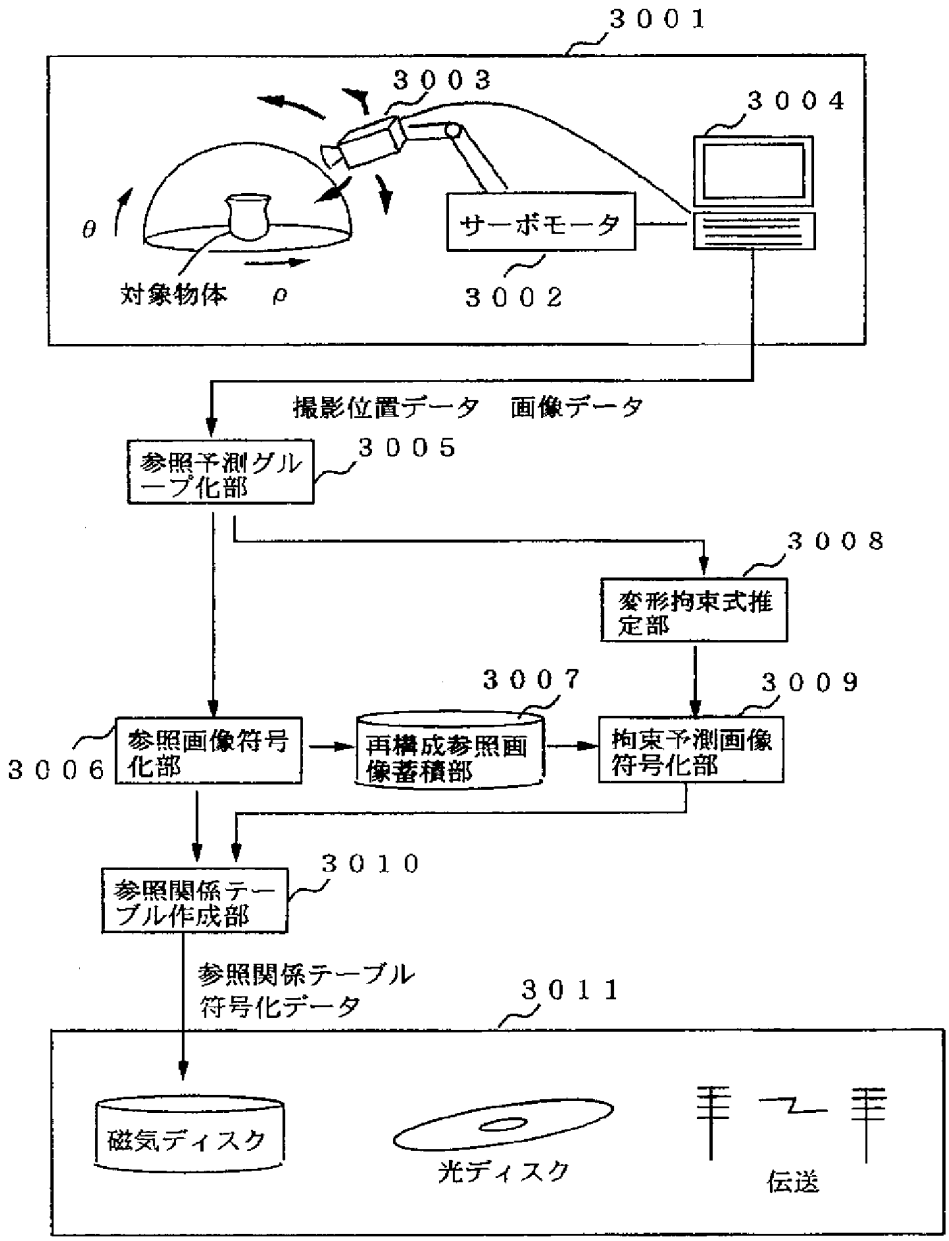
【図２４】


2401
2402
磁気ディスク
光ディスク
伝送
2403
画像要求蓄積部
2404
前回画像要求蓄積部
画像要求
ランダムアクセステーブル
符号化データ
2413
再構成参照画像群蓄積部
2414
再構成予測画像群蓄積部
2405
符号化データ選択部
2409
予測符号化データ選択部
2406
参照予測画像復号化部１
2407
参照画像復号化部１
2408
予測画像復号化部１
参照予測画像復号化部２
2410
参照画像復号化部２
2411
予測画像復号化部２
2412
2415

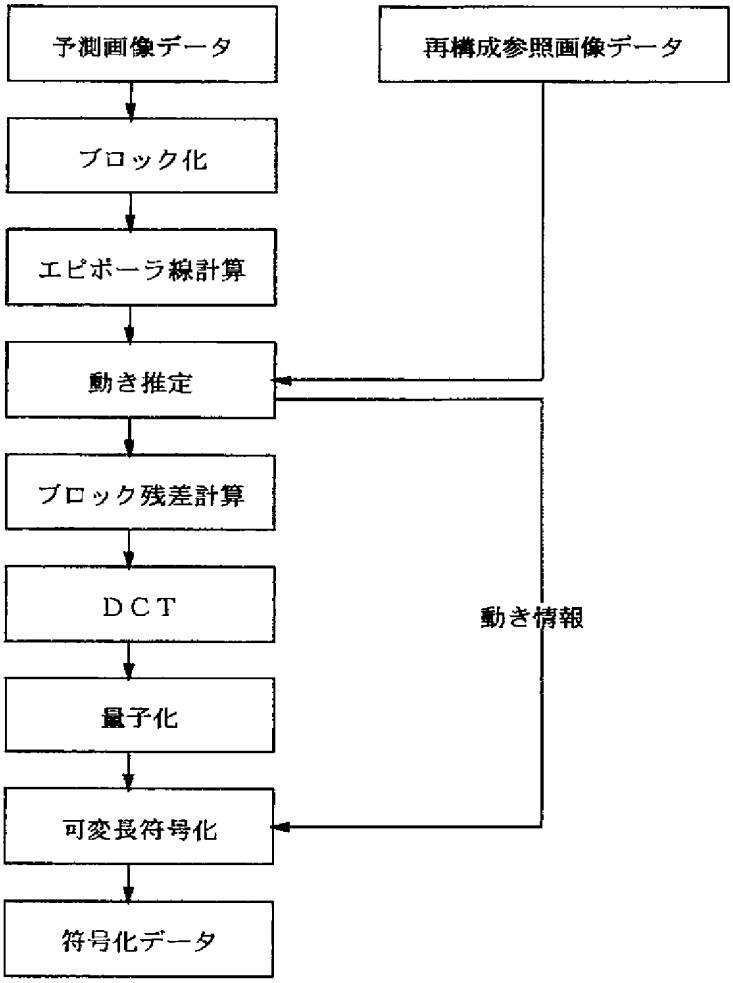
【図４１】


インデックス
参照／予測フラグ
光源位置
撮影位置
画像データ
画像データ
画像データ
画像データ

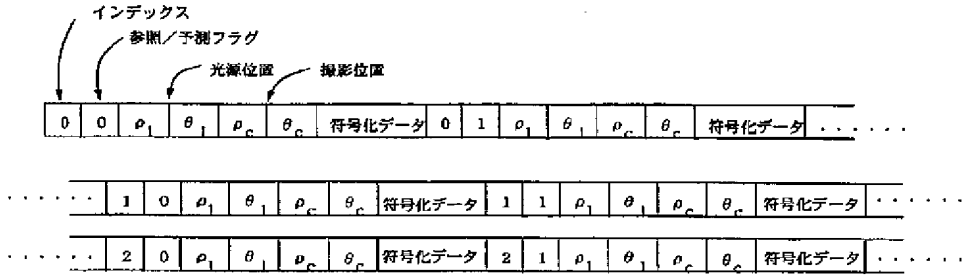
【図３０】


３００１
３００３
３００４
θ
サーボモータ
対象物体　ρ
３００２
撮影位置データ　画像データ
参照予測グループ化部
３００５
３００８
変形拘束式推定部
３００７
３００９
３００６
参照画像符号化部
再構成参照画像蓄積部
拘束予測画像符号化部
３０１０
参照関係テーブル作成部
参照関係テーブル符号化データ
３０１１
磁気ディスク
光ディスク
伝送

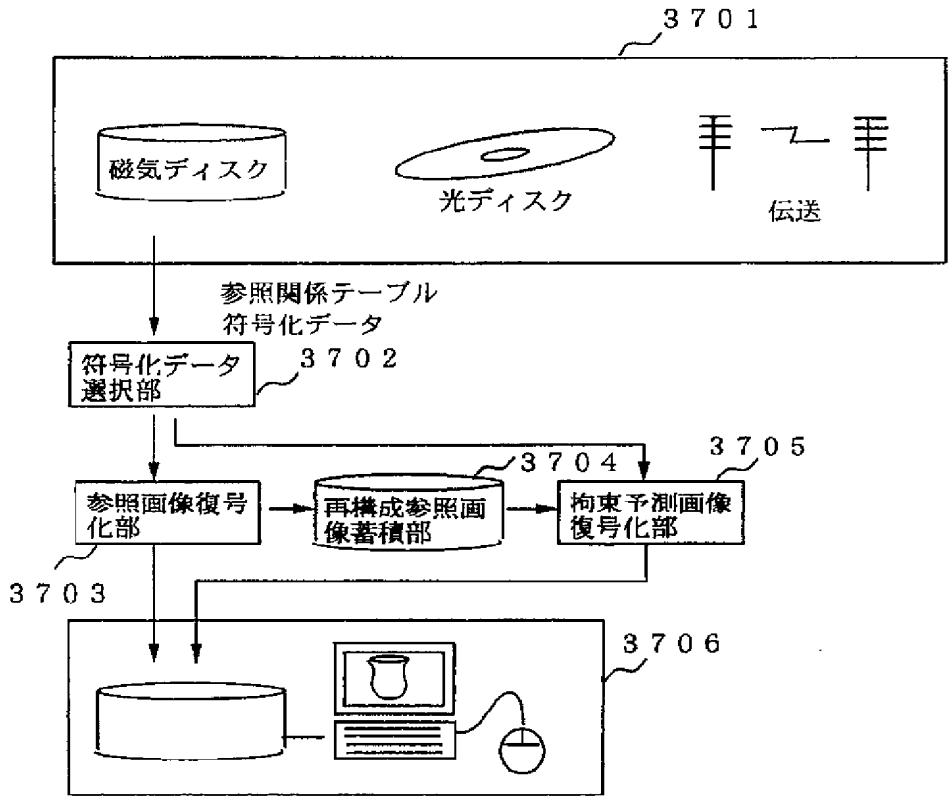
【図３４】


予測画像データ
再構成参照画像データ
ブロック化
エピポーラ線計算
動き推定
ブロック残差計算
ＤＣＴ
量子化
可変長符号化
符号化データ
動き情報

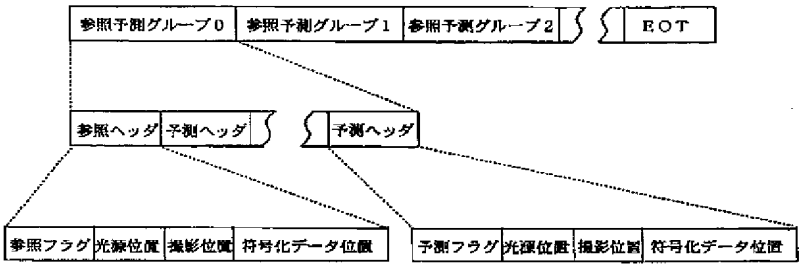
【図４２】


インデックス
参照／予測フラグ
光源位置
撮影位置
符号化データ
符号化データ
符号化データ
符号化データ
符号化データ
符号化データ

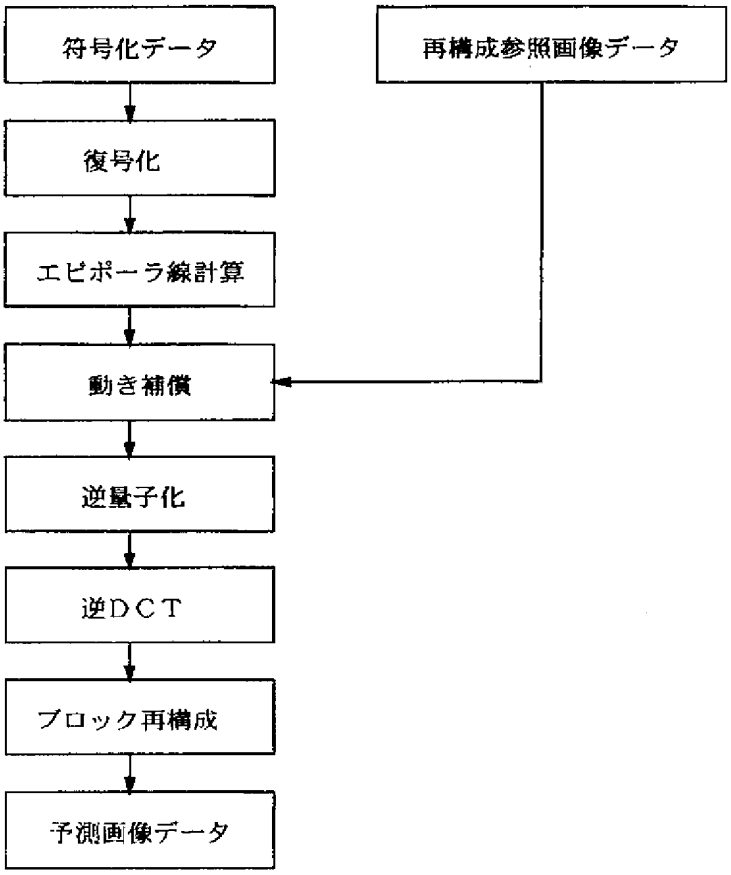
【図３７】


３７０１
磁気ディスク
光ディスク
伝送
参照関係テーブル
符号化データ
符号化データ選択部
３７０２
３７０４
３７０５
参照画像復号化部
再構成参照画像蓄積部
拘束予測画像復号化部
３７０３
３７０６

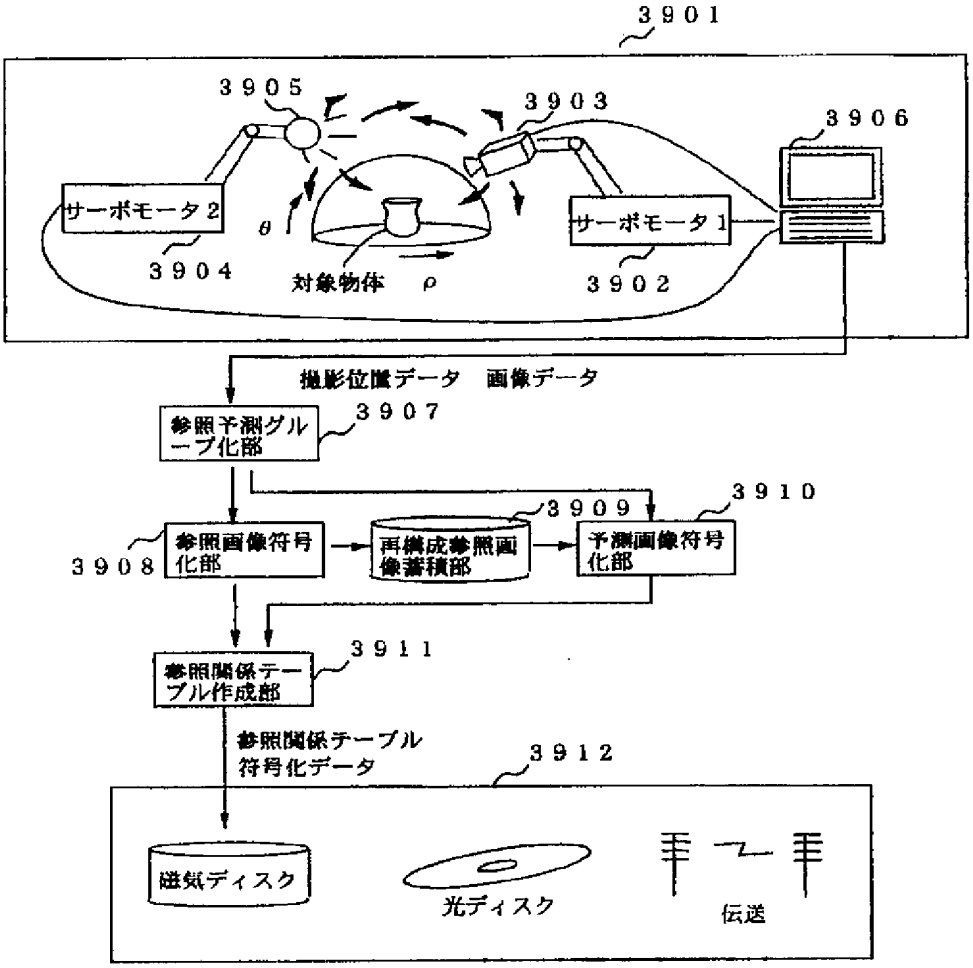
【図４３】


参照予測グループ０
参照予測グループ１
参照予測グループ２
ＥＯＴ
参照ヘッダ
予測ヘッダ
予測ヘッダ
参照フラグ
光源位置
撮影位置
符号化データ位置
予測フラグ
光源位置
撮影位置
符号化データ位置

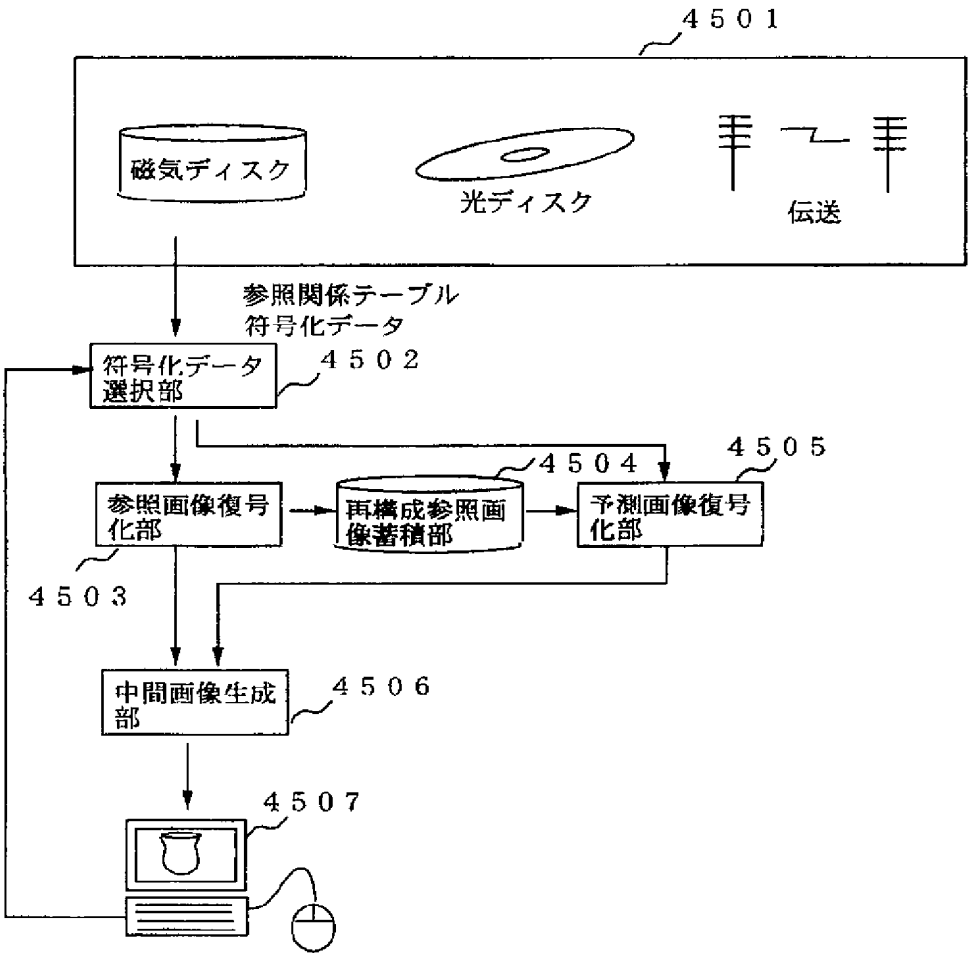
【図３８】


符号化データ
再構成参照画像データ
復号化
エピポーラ線計算
動き補償
逆量子化
逆ＤＣＴ
ブロック再構成
予測画像データ

【図３９】


３９０１
３９０５
３９０３
３９０６
サーボモータ２
θ
サーボモータ１
３９０４
対象物体
ρ
３９０２
撮影位置データ　画像データ
参照予測グループ化部
３９０７
３９１０
３９０９
３９０８
参照画像符号化部
再構成参照画像蓄積部
予測画像符号化部
３９１１
参照関係テーブル作成部
参照関係テーブル符号化データ
３９１２
磁気ディスク
光ディスク
伝送

【図４５】


４５０１
磁気ディスク
光ディスク
伝送
参照関係テーブル
符号化データ
符号化データ選択部
４５０２
４５０４
４５０５
参照画像復号化部
再構成参照画像蓄積部
予測画像復号化部
４５０３
中間画像生成部
４５０６
４５０７

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,996	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

The Japanese reference was first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), and the 05 August 2003 Official Gazette Notice, only a copy of the foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided. The U.S. patent (No. 6,266,158) is included as an English-language version of the cited Japanese reference. Also attached is a copy of the Japanese Office Action with an English translation thereof.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

By: /Erin M. Nichols/
Erin M. Nichols
Reg. No. 57,125

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	11-177987	02.07.1999	JP			Abstract	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	2010	English Translation of the Office Action of parallel Japanese application no. 2006-120296 dated 01/06/2010.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-177987
(43)Date of publication of application : 02.07.1999

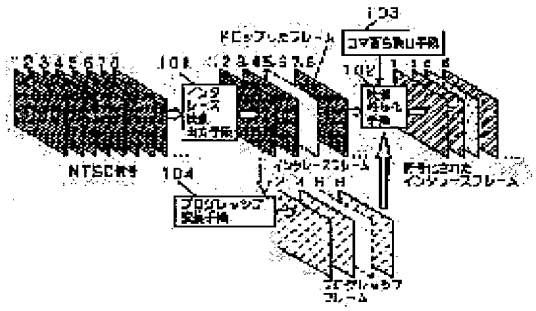
(51)Int.Cl. H04N 7/32

(21)Application number : 09-339675
(22)Date of filing : 10.12.1997

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD
(72)Inventor : YONEDA AKI
MATSUMOTO TAKAO

(54) VIDEO INFORMATION PROCESSING DEVICE AND METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To eliminate flickering or fluctuations of images caused when frame dropping occurs in an interlace frame processing and the dropped frame is supplemented by copying a frame or field immediately preceding or following the dropped frame.
SOLUTION: An interlace frame is converted once into a progressive frame 503 which is processed as an interlace frame 501. Thus, half of the time component of the interlace frame is discarded to prevent flickering of a frame-thinned part. The interlace frame is not always converted into a progressive frame but converted only when frame thinning occurs. As a result, the time component loss can be minimized and also computer resources are saved to deal with similar problems.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平11－177987

(43)公開日　平成11年(1999)７月２日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｈ０４Ｎ　7/32		Ｈ０４Ｎ　7/137	Ｚ

審査請求　未請求　請求項の数10　ＯＬ　（全 14 頁）

(21)出願番号　特願平9－339675

(22)出願日　平成９年(1997)12月10日

(71)出願人　000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地

(72)発明者　米田　亜旗
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(72)発明者　松本　孝夫
大阪府門真市大字門真1006番地　松下電器産業株式会社内

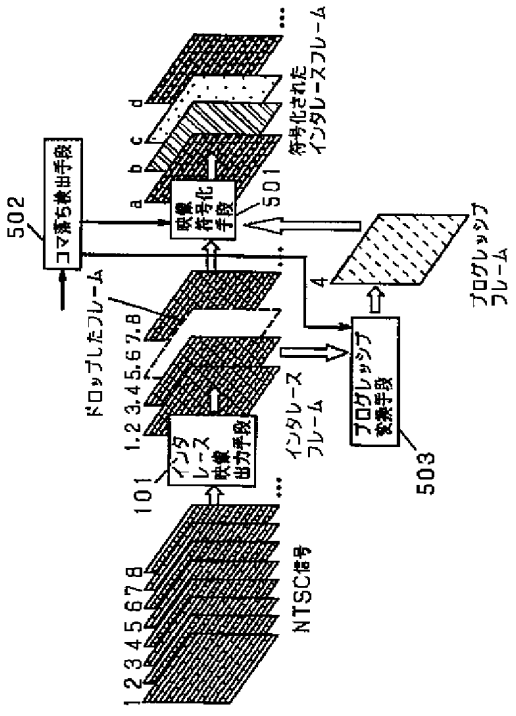
(74)代理人　弁理士　滝本　智之　（外１名）

(54)【発明の名称】　映像情報処理装置及び方法

(57)【要約】

【課題】　インタレースフレームを処理する際に、コマ落ちが発生し、そのコマ落ちを直前あるいは直後のフレームあるいはフィールドをコピーして補う場合に発生する「ちらつき」や「ゆれ」といった問題を解決することを目的とする。

【解決手段】　インタレースフレームを一旦プログレッシブフレームに変換し、そのプログレッシブフレームをインタレースフレームとして処理することで、インタレースフレームが持つ時間成分を半分廃棄し、コマ落ち部分の「ちらつき」を防ぐ。また、常にプログレッシブフレームに変換するのではなく、コマ落ちが発生した時にのみ変換することで、時間成分の損失を最小にとどめ且つ計算機資源を節約して同様の問題に対処する。



1

【特許請求の範囲】

【請求項１】n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力手段と、上記n枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換手段と、上記プログレッシブフレームのコマ落ちを検出するコマ落ち検出手段と、上記プログレッシブフレームを、n枚のフィールドから構成されるインタレースフレームとして符号化し、映像符号を出力する符号化手段であり、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ち部分にはコマ落ち部分の直前あるいは直後の上記プログレッシブフレームと同一の画像であるとする符号を出力し映像符号とする映像符号化手段とを備えたことを特徴とする映像情報処理装置。

【請求項２】n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力手段と、上記インタレースフレームのコマ落ちを検出するコマ落ち検出手段と、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを、上記インタレースフレームを構成するn枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換手段と、上記インタレース映像出力手段が出力したインタレースフレームを符号化し、映像符号を出力する符号化手段であり、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームに代わり対応する上記プログレッシブフレームをn枚のフィールドから構成されるインタレースフレームとして符号化し、且つコマ落ち部分には上記プログレッシブフレームと同一の画像であるとする符号を出力し映像符号とする映像符号化手段とを備えたことを特徴とする映像情報処理装置。

【請求項３】n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力手段と、上記インタレースフレームのコマ落ちを検出するコマ落ち検出手段と、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを構成するn枚のフィールドのうち、時間的にコマ落ち部に近い一枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換した画像を示すダミーフレーム用符号を出力するダミー用符号出力手段と、上記インタレース映像出力手段が出力したインタレースフレームを符号化し、映像符号を出力する符号化手段であり、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ち部分に出力する符号として、上記ダミーフレーム用符号を出力することで映像符号とする映像符号化手段とを備えたことを特徴とする映像情報処理装置。

【請求項４】n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力手段と、

2

上記n枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換手段と、上記プログレッシブフレームのコマ落ちを検出するコマ落ち検出手段と、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ち部の画像として、コマ落ち部の直前あるいは直後の上記プログレッシブフレームをコピーして補うコマ落ち対処手段と、上記プログレッシブ変換手段及びコマ落ち対処手段とが出力したフレームを、インタレースフレームとして、画像処理を行う画像処理手段とを備えたことを特徴とする映像情報処理装置。

【請求項５】n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力手段と、上記インタレースフレームのコマ落ちを検出するコマ落ち検出手段と、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを構成するn枚のフィールドのうち、時間的にコマ落ち部に近い一枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換し、コマ落ち部の画像を該画像で補うコマ落ち対処手段と、上記インタレース映像出力手段及びコマ落ち対処手段とが出力したフレームを、インタレースフレームとして、画像処理を行う画像処理手段とを備えたことを特徴とする映像情報処理装置。

【請求項６】n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力ステップと、上記n枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換ステップと、上記プログレッシブフレームのコマ落ちを検出するコマ落ち検出ステップと、上記プログレッシブフレームを、n枚のフィールドから構成されるインタレースフレームとして符号化し、映像符号を出力する符号化ステップであり、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ち部分にはコマ落ち部分の直前あるいは直後の上記プログレッシブフレームと同一の画像であるとする符号を出力し映像符号とする映像符号化ステップとを備えたことを特徴とする映像情報処理方法。

【請求項７】n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力ステップと、上記インタレースフレームのコマ落ちを検出するコマ落ち検出ステップと、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを、上記インタレースフレームを構成するn枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換ステップと、上記インタレース映像出力ステップが出力するインタレースフレームを符号化し、映像符号を出力する符号化ステップであり、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ

落ちの直前あるいは直後の上記インタレースフレームに代わり対応する上記プログレッシブフレームをn枚のフィールドから構成されるインタレースフレームとして符号化し、且つコマ落ち部分には上記プログレッシブフレームと同一の画像であるとする符号を出力し映像符号とする映像符号化ステップとを備えたことを特徴とする映像情報処理方法。

【請求項８】n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力ステップと、上記インタレースフレームのコマ落ちを検出するコマ落ち検出ステップと、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを構成するn枚のフィールドのうち、時間的にコマ落ち部に近い一枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換した画像を示すダミーフレーム用符号を出力するダミー用符号出力ステップと、上記インタレース映像を符号化し、映像符号を出力する符号化ステップであり、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ち部分に出力する符号として、上記ダミーフレーム用符号を出力することで映像符号とする映像符号化ステップとを備えたことを特徴とする映像情報処理方法。

【請求項９】n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力ステップと、上記n枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換ステップと、上記プログレッシブフレームのコマ落ちを検出するコマ落ち検出ステップと、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ち部の画像として、コマ落ち部の直前あるいは直後の上記プログレッシブフレームをコピーして補うコマ落ち対処ステップと、上記プログレッシブ変換ステップ及びコマ落ち対処ステップとが出力したフレームを、インタレースフレームとして、画像処理を行う画像処理ステップとを備えたことを特徴とする映像情報処理方法。

【請求項１０】n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力ステップと、上記インタレースフレームのコマ落ちを検出するコマ落ち検出ステップと、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを構成するn枚のフィールドのうち、時間的にコマ落ち部に近い一枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換し、コマ落ち部の画像を該画像で補うコマ落ち対処ステップと、上記インタレース映像出力ステップ及びコマ落ち対処ステップとが出力したフレームを、インタレースフレームとして、画像処理を行う画像処理ステップとを備えたことを特徴とする映像情報処理方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は映像情報を符号化する映像情報処理装置に構成に関する。

【０００２】

【従来の技術】近年のデジタル技術の発達により、従来ではアナログの形でしか扱えなかった、ビデオ映像のような映像情報も、デジタルの形で扱えるようになってきた。また、パーソナルコンピュータの高性能化により、デジタル化された映像情報を、パーソナルコンピュータによって処理できるようになってきた。

【０００３】NTSCのようなアナログ映像信号を、一般に販売されているパーソナルコンピュータ用の画像取り込み装置(以下、キャプチャーカード)でデジタルデータに変換したときのデータ構造を、図11で説明する。NTSC信号では、映像は偶数フィールドと、奇数フィールドで構成されている。偶数フィールドと奇数フィールドは、空間的には1ライン分のずれがあり、また時間的には、６０分の１秒分のずれがある。このようなNTSC信号の偶数フィールドと奇数フィールドを、通常のキャプチャーカードでは、奇数フィールドと偶数フィールドがライン毎に組み合わさった、1101のような１枚のフレームとして、３０分の１秒に一枚の間隔で、コンピュータ内に取り込む。このように、本来時間的にも空間的にもずれている二枚のフィールドを、一枚のフレームとして取り扱うとき、そのフレームを、インタレースフレームという。すなわち、現在普及しているキャプチャーカードでは、NTSC信号を、インタレースフレームとしてデジタル化してコンピュータ内に取り込む。

【０００４】上記したようなキャプチャーカードを用い、コンピュータ内に取り込んだインタレースフレームを符号化し、符号化映像情報として出力する映像情報処理装置の従来技術による例を、図を用いて説明する。図12は、従来技術による映像情報処理装置の構成を説明する図である。1201は、インターレース映像出力手段であり、インタレースフレームを出力するもので、一般にはキャプチャーカード等で実現され、NTSC信号を入力し、デジタル化した後にインタレースフレームとして出力するものである。1202は映像符号化手段であり、インタレース映像入力手段(1201)から入力されたインタレースフレームを、符号化し、符号化映像情報とするものである。本例では、インタレースフレームを、そのままインタレースフレームとして、符号化するものとする。例えば、MPEG2(Moving Picture ExpertGroup)の符号化規格によれば、符号化された映像情報が、インタレースフレームであることを示す符号が用意されているので、このような符号化手段を本例は想定している。しかし、それ以外の、例えば、インタレースフレームをフィールドに分割し、フィールド毎に符号化するような符号化手段を適用したとしても、従来技術を説明できる。このことは後に説明する。図13は、従来技術による映像情報処理

装置の動作を説明する図である。

【０００５】上記のように構成された従来技術による映像情報処理装置を、図12と図13を用いて説明する。まず、NTSC信号がインタレース映像出力手段(1201)に入力され、インタレース映像出力手段(1201)は、これをデジタル化し、インタレースフレームとして出力する。映像符号化手段(1202)は、上記インタレースフレームを符号化し、符号化されたインタレースフレームとして、出力する。これを繰り返す。

【０００６】上記のように出力された符号化されたインタレースフレームが復号化される様子を説明するのが、図14である。符号化されたインタレースフレームは、映像復号化手段によって、インタレースフレームに復元される。このインタレーストフレームを、NTSC出力手段が、偶数フィールドと奇数フィールドとに分割し、アナログ信号にして出力する。

【０００７】

【発明が解決しようとする課題】従来技術による映像情報処理装置は、前述の例で説明したように、インタレースフレームを、そのまま符号化するものであった。このような方式は、インタレース映像出力手段が、確実に、すべてのフレームを出力することが条件となる。もし、インタレース映像出力手段が、コマ落としなどをしてしまい、それを符号化しなければならない場合には、従来技術では本質的な問題を有している。

【０００８】図15は、上記の問題を説明するための図である。図15では、６枚のフィールドで構成されるNTSC信号がインタレース映像出力手段に入力されたものとする。６枚のフィールドの時間的な並びを示すため、1～6の番号を示した。1がもっとも過去の映像であり、6がもっとも新しい映像である。上記のごとく６枚のフィールド(1～6)が、インタレース映像出力手段に入力されたが、符号化の負荷が高く、デジタル化したインタレース映像のバッファが足らなくなった、等の理由により、コマ落ちし、６枚のフィールドのうち、中心の二枚(3,4)がインタレースフレームにならなかったものとする。すなわち、本来なら３枚のインタレースフレームが出力されるところが、２枚しか出力されなかった。そこで、映像符号化手段は、コマ落ちしたフレームに対して、「前回のフレームと同一」という符号を出力し(図15のb)、符号化映像情報とした。

【０００９】上記のごとく符号化された映像情報を復号化される様子を説明するのが、図16である。上記で符号化された映像情報は、映像復号化手段によりインタレースフレームに復号化されるが、符号化映像のbは、「前回のフレームと同一」という符号なので、最初の一枚のインタレースフレームが、コピーされ、二枚目のインタレースフレームとして使用される。このように復号化されたインタレースフレームを、NTSC出力手段が、偶数フィールドと奇数フィールドとに分割し、アナログ信号にして出力する。この時、最初のインタレースフレームは、「1,2」という時間的に変化するフィールドを持っており、２枚目のフレームは、最初のフレームを、コピーして使用するので、二枚目のフレームも、「1,2」と時間的に変化するフィールドを持つこととなる。これをNTSC信号として出力すると、最初、「1,2」とフィールドを出力した後に、もう一度、「1」のフィールドを出力する。これは、瞬間的にだが、「過去に戻る」こととなり、動きのある映像を出力すると、「1,2」と動いた映像が「1」で戻るので、映像が「ちらつく」という問題が発生してしまう。

【００１０】図16で示した問題を回避するため、図15におけるbを、「偶数、奇数フィールドとも、前回の偶数フィールドと同じ」という符号とする。すなわち、図16で示した問題は、そもそも時間的に異なる２つのフィールドを持つフレームを、そのままコピーして使用したことに起因するものであるから、単一のフィールドをコピーして使用すれば、問題は回避できるはずである。

【００１１】上記のごとく符号化された映像情報を復号化される様子を説明するのが、図17である。上記で符号化された映像情報は、映像復号化手段によりインタレースフレームに復号化されるが、符号化映像のbは、「偶数、奇数フィールドとも、前回の偶数フィールドと同じ」という符号なので、最初の一枚のインタレースフレームのうち、偶数フィールドのみコピーされ、二枚目のインタレースフレームとして使用される。このように復号化されたインタレースフレームを、NTSC出力手段が、偶数フィールドと奇数フィールドとに分割し、アナログ信号にして出力する。この時、２枚目のフレームは、最初のフレームのうち、偶数フィールドを二回コピーして使用するので、二枚目のフレームは、「2,2」と、時間的に変化しない、同一のフィールドを持つこととなる。これをNTSC信号として出力すると、時間的には、「1,2,2,2,5,6」となり、「過去に戻る」ことはなくなるので、図16のように、「ちらつく」という問題は回避できる。しかし、空間的には、本来、偶数フィールドにある画像を、奇数フィールド、すなわち１ライン上に表示することになるので(図17のX)、画像が上下に「ゆれる」という問題が発生する。これは、静止している映像を出力すると、特に顕著に知覚される問題である。

【００１２】上記した従来の技術による例では、映像符号化手段としては、インタレースフレームを、そのままインタレースフレームとして、符号化するものとしたが、それ以外の、例えば、インタレースフレームをフィールドに分割し、フィールド毎に符号化するような符号化手段を適用したとしても、上記と同様の問題が発生することは、いうまでもない。また、本例では、コマ落ちした部分に対して、「前回のフレームと同じ」あるいは「前回のフィールドと同じ」という符号を用いるものとしたが、符号を用いず、前回のフレーム、あるいはフィ

ールドと同じ映像をそのまま用いてコマ落ちした部分を符号化しても、当然ながら同様の問題が発生する。また、「前回のフレームと同じ」という符号ではなく、「次回のフレームと同じ」という符号を用いたとしても、やはり、同様の問題が発生する。すなわち、コマ落ちした部分を補うのに使用するフィールドが、コマ落ちした部分のフィールドと、偶数、奇数が異なれば、空間的な問題が発生し、偶数、奇数をあわせようとすると、時間的な問題が発生するのである。　本発明は、従来の映像情報処理装置が有していた前記の問題に鑑み、システムの負荷変動によりコマ落ちが発生しても、「ちらつき」や「ゆれ」といった問題を回避しつつ、コマ落ち部分を補うことを可能とすることを目的とするものである。

【００１３】

【課題を解決するための手段】上記目的を達成するために、請求項１、及び請求項４にかかる映像情報処理装置は、すべてのインタレースフレームを、一旦、フレームを構成するフィールドのうちの一枚をプログレッシブ変換することでプログレッシブフレームにし、そのプログレッシブフレームをインタレースフレームとして処理し、コマ落ち部にはコマ落ちの直前あるいは直後のフレームをコピーして補うことを特徴とするものである。

【００１４】また、請求項２にかかる映像情報処理装置は、コマ落ちがあった場合にのみ、コマ落ちの直前あるいは直後のインタレースフレームを、一旦、フレームを構成するフィールドのうちの一枚をプログレッシブ変換することでプログレッシブフレームにし、そのプログレッシブフレームをインタレースフレームとして処理し、コマ落ち部には、コマ落ちの直前あるいは直後のフレームをコピーして補うことを特徴とするものである。

【００１５】また、請求項３及び請求項５にかかる映像情報処理装置は、コマ落ちがあった場合にのみ、コマ落ちの直前あるいは直後のフィールドをプログレッシブ変換することでプログレッシブフレームにし、コマ落ち部には、このプログレッシブフレームをインタレースフレームとして処理することで補うことを特徴とするものである。

【００１６】また、請求項６、及び請求項９にかかる映像情報処理方法は、すべてのインタレースフレームを、一旦、フレームを構成するフィールドのうちの一枚をプログレッシブ変換することでプログレッシブフレームにし、そのプログレッシブフレームをインタレースフレームとして処理し、コマ落ち部にはコマ落ちの直前あるいは直後のフレームをコピーして補うことを特徴とするものである。

【００１７】また、請求項７にかかる映像情報処理方法は、コマ落ちがあった場合にのみ、コマ落ちの直前あるいは直後のインタレースフレームを、一旦、フレームを構成するフィールドのうちの一枚をプログレッシブ変換

することでプログレッシブフレームにし、そのプログレッシブフレームをインタレースフレームとして処理し、コマ落ち部には、コマ落ちの直前あるいは直後のフレームをコピーして補うことを特徴とするものである。

【００１８】また、請求項８及び請求項１０にかかる映像情報処理方法は、コマ落ちがあった場合にのみ、コマ落ちの直前あるいは直後のフィールドをプログレッシブ変換することでプログレッシブフレームにし、コマ落ち部には、このプログレッシブフレームをインタレースフレームとして処理することで補うことを特徴とするものである。

【００１９】

【発明の実施の形態】（実施の形態１）本発明の実施の形態１による映像情報処理装置は、すべてのインタレースフレームのフィールドを一旦プログレッシブフレームに変換し、それをインタレースフレームとして符号化するものである。

【００２０】図1は、本発明の第１の実施の形態における映像情報処理装置の構成図であり、図2は、同装置の動作を説明する図である。図中、101はインターレース映像出力手段であり、インターレースフレームを出力するもので、一般にはキャプチャーカード等で実現され、NTSC信号を入力し、デジタル化した後にインタレースフレームとして出力するものである。102は映像符号化手段であり、プログレッシブ変換手段(104)が変換したプログレッシブフレームを、インタレースフレームとして、符号化するものである。また、コマ落ち検出手段が(103)がプログレッシブ変換手段が出力したプログレッシブフレームのコマ落ちを検出すると、コマ落ち部分に、「コマ落ちの直前のインタレースフレームと同一」という符号を出力するものである。103はコマ落ち検出手段であり、プログレッシブ変換手段(104)が出力するプログレッシブフレームのコマ落ちを検出するものである。104は、プログレッシブ変換手段であり、インタレース映像出力手段(101)が出力したインタレースフレームを構成するフィールドのうち、一枚のフィールドを、プログレッシブフレームに変換するものである。ここで、フィールドをプログレッシブフレームに変換する様子を、図3で示す。図3では、フィールド２の、間隙となっているラインを補間することで、プログレッシブフレームに変換している。フィールド２のピクセルａは、隣あう上下のラインを構成するピクセルの平均をとることで作成している。これをフィールド内補間という。また、フィールド２のピクセルｂは、先立つフィールド１から、対応する位置のピクセルをコピーすることで作成している。これをフィールド間補間という。フィールド間補間をすれば厳密にプログレッシブフレームの解像度が倍になるが、動きのある部分でフィールド間補間をすると、像が２重となり、見ずらい画像となる。フィールド内補間の場合は、解像度は落ちるが、動きのある部分

でも適応できる。このように、フィールドをライン補間してプログレッシブフレームにする技術については、多くの研究がなされている。本例のプログレッシブ変換手段(104)は、いずれかの方法を用いて、インタレースフレームを構成するフィールドのうち、偶数フィールドを、プログレッシブフレームに変換するものとする。上記の例でもわかるように、変換されたプログレッシブフレームは、垂直解像度は倍になっているが、時間的には、偶数フィールドの時点での時間成分のみを持つものとなる。

【００２１】上記のように構成された第１の実施の形態における映像情報処理装置の動作を、図1と図2を用いて説明する。まず、NTSC信号がインタレース映像出力手段(101)に入力され、インタレース映像出力手段(101)は、これをデジタル化し、インタレースフレームとして出力する。ここでは、「1,2,3,4,5,6,7,8」の順序でフィールドが入力されたが、コマ落ちが起こり、本来ならば4枚出力されるはずのインタレースフレームが、「1,2」と、「3,4」と、「7,8」の時間要素を持つ三枚しか、出力されなかった。このインタレースフレームを、プログレッシブ変換手段(104)が、偶数フィールドのみ、プログレッシブフレームにする。偶数フィールドのみをプログレッシブフレームにするので、垂直解像度は倍だが、時間要素は「2」と「4」と「8」である、プログレッシブフレームが３枚出力される。このプログレッシブフレームを、映像符号化手段(102)が、インターレースフレームとして、符号化し、「a」と「b」と「d」を得る。また、コマ落ち検出手段(103)は、プログレッシブフレームに付与されたタイムスタンプを読み取ることで、コマ落ちが発生したことを検出する。コマ落ち検出手段(103)からの指示により、映像符号化手段(102)は、「b」と同一であるという、「c」という符号を出力する。

【００２２】上記のように出力された符号化されたインタレースフレームが復号化される様子を説明するのが、図4である。符号化されたインタレースフレームは、映像復号化手段によって、インタレースフレームに復元される。このインタレーストフレームを、NTSC出力手段が、偶数フィールドと奇数フィールドとに分割し、アナログ信号にして出力する。このとき、「c」は、「b」と同じであるという符号になっており、また、「a」と「b」と「d」は、一度プログレッシブフレームに変換した画像を、インタレースフレームとして符号化したものであるから、垂直解像度はフィールドの倍になっているものの、時間的要素は「2」と「4」と「8」しかない。したがって、再生される映像の時間的要素は、「2,2,4,4,4,4,8,8」となる。すなわち、従来の技術で問題となった、「時間が戻る」ことによる「ちらつき」の問題が解決でき、さらに、プログレッシブフレームに変換することで垂直解像度を補間しているので、本来偶数フィールドで表示すべき画像を奇数フィールドで表示することによる「ゆれ」の問題も解決できる。

【００２３】なお、本例では、偶数フィールドをプログレッシブ変換するものとしたが、これは、奇数フィールドをプログレッシブ変換することもできる。また、本例では、コマ落ち検出手段が、プログレッシブフレームに付与されたタイムスタンプを読み取ることでコマ落ちを検出するものとしたが、プログレッシブ変換手段、あるいはインタレース映像出力手段から、コマ落ちした情報を得ても実現できる。また、本例では、映像符号化手段としては、プログレッシブフレームを、インタレースフレームとして、符号化するものとしたが、それ以外の、例えば、プログレッシブフレームをライン毎に分割してフィールドとし、フィールド毎に符号化するような符号化手段を適用したとしても、本発明は適用できる。また、本例では、コマ落ちした部分に対して、「前回のフレームと同じ」という符号を用いるものとしたが、符号を用いず、前回のフレーム、と同じ映像をそのまま用いてコマ落ちした部分を符号化しても、当然ながら問題は解決できる。また、「前回のフレームと同じ」という符号ではなく、「次回のフレームと同じ」という符号を用いたとしても、やはり、問題は解決できる。

【００２４】さらに、本実施の形態では、出力されたインタレースフレームの処理として、画像の符号化を行うものとしたが、本発明はこのような処理に限定されるものではない。符号化処理ではなく、一般的な画像処理においても、本実施の形態で示したような本発明の実施が行えることはいうまでもない。その場合、コマ落ち部には、「直前、あるいは直後と同一」という符号を用いるのではなく、直前、あるいは直後のフレームをコピーし、コマ落ちを補う、コマ落ち対処手段、等を用意すればよい。

【００２５】（実施の形態２）本発明の実施の形態２による映像情報処理装置は、コマ落ちがあった時に、コマ落ちの直前あるいは直後のインタレースフレームのみ、フィールドを一旦プログレッシブフレームに変換し、それをインタレースフレームとして符号化し、且つ、コマ落ち部分には、プログレッシブフレームに変換した画像と同一であるという符号を出力するものでる。

【００２６】図5は、本発明の第２の実施の形態における映像情報処理装置の構成図であり、図6は、同装置の動作を説明する図である。図中、501は映像符号化手段であり、インタレース映像出力手段(101)が出力したインタレースフレームを符号化するものであり、且つ、コマ落ち検出部(502)がインタレースフレームのコマ落ちを検出すると、コマ落ちの直前のインタレースフレームに関してはこれを符号化せず、代わりにプログレッシブ変換手段(503)が出力したプログレッシブフレームを、インタレースフレームとして、符号化し、さらに、コマ落ち部分に関しては、「直前のインタレースフレームと同一」という符号を出力するものである。502はコマ落

ち検出手段であり、インタレース映像出力手段(101)が出力したインタレースフレームのコマ落ちを検出するものである。503は、プログレッシブ変換手段であり、コマ落ち検出手段(502)がコマ落ちを検出すると、コマ落ち部分の直前のインタレースフレームを構成するフィールドのうち、偶数フィールドを、プログレッシブフレームに変換し、出力するものである。フィールドをプログレッシブフレームに変換する方法については、実施の形態１で説明した方法と同じである。その他の構成要素については実施の形態１と同様であるので、説明は割愛する。

【００２７】上記のように構成された第２の実施の形態における映像情報処理装置の動作を、図5と図6を用いて説明する。まず、NTSC信号がインタレース映像出力手段(101)に入力され、インタレース映像出力手段(101)は、これをデジタル化し、インタレースフレームとして出力する。ここでは、「1,2,3,4,5,6,7,8」の順序でフィールドが入力されたが、コマ落ちが起こり、本来ならば４枚出力されるはずのインタレースフレームが、「1,2」と、「3,4」と、「7,8」の時間要素を持つ三枚しか、出力されなかった。このコマ落ちを、コマ落ち検出手段(502)が、インタレースフレームに付与されたタイムスタンプを読み取ることで検出し、プログレッシブ変換手段(503)に対して、変換の指示を出す。プログレッシブ変換手段(503)は、変換の指示を受けて、コマ落ち部分の直前のインタレースフレームの、偶数フィールドのみ、プログレッシブフレームにする。偶数フィールドのみをプログレッシブフレームにするので、垂直解像度は倍だが、時間要素は「4」である、プログレッシブフレームが一枚出力される。一方、映像符号化手段(501)は、コマ落ち検出手段(502)のコマ落ちの指示を受け、コマ落ち部分の直前のインタレースフレーム(3,4)に関しては符号化せず、代わりにプログレッシブ変換手段(503)が出力したプログレッシブフレームを、インタレースフレームとして、符号化する(図中の「b」)。また、コマ落ち部分に関しては、「直前のインタレースフレームと同一」という符号(図中の「c」)を出力する。さらに、その後のインタレースフレーム(7,8)については、通常どおり、インタレースフレームとして、符号化する(図中の「d」)。

【００２８】上記のように出力された符号化されたインタレースフレームが復号化される様子を説明するのが、図7である。符号化されたインタレースフレームは、映像復号化手段によって、インタレースフレームに復元される。このインタレーストフレームを、NTSC出力手段が、偶数フィールドと奇数フィールドとに分割し、アナログ信号にして出力する。このとき、「c」は、「b」と同じであるという符号になっており、また、「b」は、一度プログレッシブフレームに変換した画像を、インタレースフレームとして符号化したものであるから、垂直解像度はフィールドの倍になっているものの、時間的要素は「4」しかない。したがって、再生される映像の時間的要素は、「1,2,4,4,4,4,7,8」となる。すなわち、従来の技術で問題となった、「ちらつき」の問題が解決でき、さらに、プログレッシブフレームに変換することで垂直解像度を補間しているので、本来偶数フィールドで表示すべき画像を奇数フィールドで表示することによる「ゆれ」の問題も解決できる。さらに、実施の形態１では、すべてのフレームの偶数フィールドをプログレッシブ変換していたため、すべてのフレームの時間的要素が半分となり、コマ落ちが発生しなかったとしても、本来1/60秒に一回動くはずの映像が、1/30秒に一回となり、動きがカクカクするという問題を有していたが、本実施の形態では、コマ落ちが発生した場合にのみ、プログレッシブ変換を行うので、コマ落ちが発生しない限り、1/60秒の滑らかな再生が実現できる。さらに、コマ落ちが生じた時のみプログレッシブ変換を行うことで、毎フレームごとにプログレッシブ変換を行う場合に比べ、計算機資源の節約が図れるという利点もある。

【００２９】なお、本例では、偶数フィールドをプログレッシブ変換するものとしたが、これは、奇数フィールドをプログレッシブ変換することもできる。また、本例では、映像符号化手段としては、プログレッシブフレームを、インタレースフレームとして、符号化するものとしたが、それ以外の、例えば、プログレッシブフレームをライン毎に分割してフィールドとし、フィールド毎に符号化するような符号化手段を適用したとしても、本発明は適用できる。また、本例では、コマ落ちした部分に対して、「前回のフレームと同じ」という符号を用いるものとしたが、符号を用いず、前回のフレーム、と同じ映像をそのまま用いてコマ落ちした部分を符号化しても、当然ながら問題は解決できる。また、コマ落ち部分の直前のインタレースフレームをプログレッシブ変換するのではなく、コマ落ち部分の直後のインタレースフレームをプログレッシブ変換し、コマ落ち部分には、「前回のフレームと同じ」という符号ではなく、「次回のフレームと同じ」という符号を用いても、問題は解決できる。

【００３０】（実施の形態３）本発明の実施の形態３による映像情報処理装置は、コマ落ちがあった時に、コマ落ちの直前あるいは直後のインタレースフレームの、コマ落ち部分に時間的に近いフィールドを、一旦プログレッシブフレームに変換し、それをインタレースフレームとして符号化して、コマ落ち部分の符号として用いるものである。

【００３１】図8は、本発明の第２の実施の形態における映像情報処理装置の構成図であり、図9は、同装置の動作を説明する図である。図中、801は映像符号化手段であり、インタレース映像出力手段(101)が出力したインタレースフレームを符号化するものであり、且つ、コ

マ落ち検出部(502)がインタレースフレームのコマ落ちを検出すると、コマ落ち部分の符号として、ダミー用符号出力手段(802)が出力したダミー用の符号を用いるものである。802は、ダミー用符号出力手段であり、コマ落ち検出部(502)がインタレースフレームのコマ落ちを検出すると、コマ落ち部分の直前のインタレースフレームを構成するフィールドのうち、コマ落ち部に時間的に近いフィールド、すなわち、この場合は偶数フィールドを、プログレッシブフレームに変換し、その画像を符号化して、出力するものである。フィールドをプログレッシブフレームに変換する方法については、実施の形態１で説明した方法と同じである。その他の構成要素については実施の形態２と同様であるので、説明は割愛する。

【００３２】上記のように構成された第３の実施の形態における映像情報処理装置の動作を、図8と図9を用いて説明する。まず、NTSC信号がインタレース映像出力手段(101)に入力され、インタレース映像出力手段(101)は、これをデジタル化し、インタレースフレームとして出力する。ここでは、「1,2,3,4,5,6,7,8」の順序でフィールドが入力されたが、コマ落ちが起こり、本来ならば４枚出力されるはずのインタレースフレームが、「1,2」と、「3,4」と、「7,8」の時間要素を持つ三枚しか、出力されなかった。このコマ落ちを、コマ落ち検出手段(502)が、インタレースフレームに付与されたタイムスタンプを読み取ることで検出し、ダミー用符号出力手段(802)に対して、ダミー用符号出力の指示を出す。ダミー用符号出力手段(802)は、上記の指示を受けて、コマ落ち部分の直前のインタレースフレームの、偶数フィールドを、プログレッシブフレームにする。偶数フィールドのみをプログレッシブフレームにするので、垂直解像度は倍だが、時間要素は「4」である。このプログレッシブフレームを、インタレースフレームとして、符号化して、出力する。一方、映像符号化手段(801)は、コマ落ち検出手段(502)のコマ落ちの指示を受け、コマ落ち部分に、ダミー用符号出力手段(802)が出力したダミー用符号を出力する(図中の「c」)。さらに、その後のインタレースフレーム(7,8)については、通常どおり、インタレースフレームとして、符号化する(図中の「d」)。

【００３３】上記のように出力された符号化されたインタレースフレームが復号化される様子を説明するのが、図10である。符号化されたインタレースフレームは、映像復号化手段によって、インタレースフレームに復元される。このインタレーストフレームを、NTSC出力手段が、偶数フィールドと奇数フィールドとに分割し、アナログ信号にして出力する。このとき、「c」は、一度プログレッシブフレームに変換した画像を、インタレースフレームとして符号化したものであるから、垂直解像度はフィールドの倍になっているものの、時間的要素は「4」しかない。したがって、再生される映像の時間的要素は、「1,2,3,4,4,4,7,8」となる。すなわち、従来

の技術で問題となった、「ちらつき」の問題が解決でき、さらに、プログレッシブフレームに変換することで垂直解像度を補間しているので、本来偶数フィールドで表示すべき画像を奇数フィールドで表示することによる「ゆれ」の問題も解決できる。さらに、実施の形態２では、コマ落ちの直前のフィールドをプログレッシブフレームに変換していたため、コマ落ち部分と合わせて多くの時間成分が失われていたという問題を有していたが、本実施の形態では、コマ落ち部分のみ時間成分が失われるので、滑らかな再生が実現できる。

【００３４】なお、本例では、ダミー用符号出力手段と映像符号化手段としては、プログレッシブフレームを、インタレースフレームとして、符号化するものとしたが、それ以外の、例えば、プログレッシブフレームをライン毎に分割してフィールドとし、フィールド毎に符号化するような符号化手段を適用したとしても、本発明は適用できる。また、本例では、コマ落ちした部分に対して、まず、コマ落ちの直前のフィールドをプログレッシブフレームに変換し、その画像をインタレースフレームとして符号化した符号を用いていたが、本発明を実装する処理系に、「前回のフィールドをプログレッシブ変換したものと同一(あるいは、前回のフィールドのライン補間したものと同一)」という符号が用意されていれば、そのような符号をそのまま用いてもよい。また、コマ落ち部分の直前のフィールドをプログレッシブ変換するのではなく、コマ落ち部分の直後のフィールドをプログレッシブ変換し、それを符号化し、コマ落ち部分に用いても、問題は解決できる。その場合、変換するフィールドは、奇数フィールドとなる。

【００３５】さらに、本実施の形態では、出力されたインタレースフレームの処理として、画像の符号化を行うものとしたが、本発明はこのような処理に限定されるものではない。符号化処理ではなく、一般的な画像処理においても、本実施の形態で示したような本発明の実施が行えることはいうまでもない。その場合、ダミー用符号出力手段の代わりに、直前、あるいは直後のフィールドをプログレッシブ変換し、その画像をコマ落ちを補う、コマ落ち対処手段、等を用意すればよい。

【００３６】さらに、これまで述べた本発明の方法を、プログラム製品として実現し、それをフロッピーディスクやその他の記憶媒体などに格納して配布した場合や、コンピュータネットワーク等の媒体を通じて配布した場合などでも、本発明が有効であることは言うまでもない。

【００３７】

【発明の効果】以上のように、本発明の請求項１にかかる映像情報処理装置は、n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力手段と、上記n枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに

変換するプログレッシブ変換手段と、上記プログレッシブフレームのコマ落ちを検出するコマ落ち検出手段と、上記プログレッシブフレームを、n枚のフィールドから構成されるインタレースフレームとして符号化し、映像符号を出力する符号化手段であり、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ち部分にはコマ落ち部分の直前あるいは直後の上記プログレッシブフレームと同一の画像であるとする符号を出力し映像符号とする映像符号化手段とを備えたことを特徴とすることで、インタレースフレームを符号化する際に、コマ落ちが発生し、そのコマ落ちを直前あるいは直後のフレームあるいはフィールドをコピーして補う場合に発生する「ちらつき」や「ゆれ」といった問題を解決することができる。

【００３８】請求項２にかかる映像情報処理装置は、n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力手段と、上記インタレースフレームのコマ落ちを検出するコマ落ち検出手段と、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを、上記インタレースフレームを構成するn枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換手段と、上記インタレースフレームを符号化し、映像符号を出力する符号化手段であり、上記コマ落ち検出手段がコマ落ちを検出すると、上記インタレースフレームに代わり対応する上記プログレッシブフレームをn枚のフィールドから構成されるインタレースフレームとして符号化し、且つコマ落ち部分には上記プログレッシブフレームと同一の画像であるとする符号を出力し映像符号とする映像符号化手段とを備えたことを特徴とすることで、コマ落ちがない部分は1/60fpsの滑らかな動きを保ちつつ、また、プログレッ　シブ変換のための計算機資源を節約した上で、請求項１で述べた効果を実現できる。

【００３９】請求項３にかかる映像情報処理装置は、n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力手段と、上記インタレースフレームのコマ落ちを検出するコマ落ち検出手段と、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを構成するn枚のフィールドのうち、時間的にコマ落ち部に近い一枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換した画像を示すダミーフレーム用符号を出力するダミー用符号出力手段と、上記インタレース映像を符号化し、映像符号を出力する符号化手段であり、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ち部分に出力する符号として、上記ダミー用符号を出力することで映像符号とする映像符号化手段とを備えたことを特徴とすることで、時間的成分の欠落をコマ落ち部分のみにとどめ、請求項１で述べた効果を実現できる。

【００４０】請求項４にかかる映像情報処理装置は、n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力手段と、上記n枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換手段と、上記プログレッシブフレームのコマ落ちを検出するコマ落ち検出手段と、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ち部の画像として、コマ落ち部の直前あるいは直後の上記プログレッシブフレームをコピーして補うコマ落ち対処手段と、上記プログレッシブ変換手段及びコマ落ち対処手段とが出力したフレームを、インタレースフレームとして、画像処理を行う画像処理手段とを備えたことを特徴とすることで、インタレースフレームを処理する際に、コマ落ちが発生し、そのコマ落ちを直前あるいは直後のフレームあるいはフィールドをコピーして補う場合に発生する「ちらつき」や「ゆれ」といった問題を解決することができる。

【００４１】請求項５にかかる映像情報処理装置は、n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力手段と、上記インタレースフレームのコマ落ちを検出するコマ落ち検出手段と、上記コマ落ち検出手段がコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを構成するn枚のフィールドのうち、時間的にコマ落ち部に近い一枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換し、コマ落ち部の画像を該画像で補うコマ落ち対処手段と、上記インタレース映像出力手段及びコマ落ち対処手段とが出力したフレームを、インタレースフレームとして、画像処理を行う画像処理手段とを備えたことを特徴とすることで、時間的成分の欠落をコマ落ち部分のみにとどめ、請求項４で述べた効果を実現できる。

【００４２】請求項６にかかる映像情報処理方法は、n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力ステップと、上記n枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換ステップと、上記プログレッシブフレームのコマ落ちを検出するコマ落ち検出ステップと、上記プログレッシブフレームを、n枚のフィールドから構成されるインタレースフレームとして符号化し、映像符号を出力する符号化ステップであり、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ち部分にはコマ落ち部分の直前あるいは直後の上記プログレッシブフレームと同一の画像であるとする符号を出力し映像符号とする映像符号化ステップとを備えたことを特徴とすることで、インタレースフレームを符号化する際に、コマ落ちが発生し、そのコマ落ちを直前あるいは直後のフレームあるいはフィールドをコピーして補う場合に発生する「ちらつき」や「ゆれ」といった問題を解決することができる。

【００４３】請求項７にかかる映像情報処理方法は、ｎ枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力ステップと、上記インタレースフレームのコマ落ちを検出するコマ落ち検出ステップと、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを、上記インタレースフレームを構成するn枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換ステップと、上記インタレースフレームを符号化し、映像符号を出力する符号化ステップであり、上記コマ落ち検出ステップがコマ落ちを検出すると、上記インタレースフレームに代わり対応する上記プログレッシブフレームをn枚のフィールドから構成されるインタレースフレームとして符号化し、且つコマ落ち部分には上記プログレッシブフレームと同一の画像であるとする符号を出力し映像符号とする映像符号化ステップとを備えたことを特徴とすることで、コマ落ちがない部分は1/60 fpsの滑らかな動きを保ちつつ、また、プログレッシブ変換のための計算機資源を節約した上で、請求項６で述べた効果を実現できる。

【００４４】請求項８にかかる映像情報処理方法は、ｎ枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力ステップと、上記インタレースフレームのコマ落ちを検出するコマ落ち検出ステップと、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを構成するn枚のフィールドのうち、時間的にコマ落ち部に近い一枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換した画像を示すダミーフレーム用符号を出力するダミー用符号出力ステップと、上記インタレース映像を符号化し、映像符号を出力する符号化ステップであり、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ち部分に出力する符号として、上記ダミー用符号を出力することで映像符号とする映像符号化ステップとを備えたことを特徴とすることで、時間的成分の欠落をコマ落ち部分のみにとどめ、請求項６で述べた効果を実現できる。

【００４５】請求項９にかかる映像情報処理方法は、ｎ枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力ステップと、上記n枚のフィールドのうち1枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換するプログレッシブ変換ステップと、上記プログレッシブフレームのコマ落ちを検出するコマ落ち検出ステップと、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ち部の画像として、コマ落ち部の直前あるいは直後の上記プログレッシブフレームをコピーして補うコマ落ち対処ステップと、上記プログレッシブ変換ステップ及びコマ落ち対処ステップとが出力したフレームを、インタレースフレームとして、画像処理を行う画像処理ステップとを備えたことを特徴とすることで、インタレースフレームを処理する際に、コマ落ちが発生し、そのコマ落ちを直前あるいは直後のフレームあるいはフィールドをコピーして補う場合に発生する「ちらつき」や「ゆれ」といった問題を解決することができる。

【００４６】請求項１０にかかる映像情報処理方法は、n枚のフィールドから構成されるインタレースフレームを出力するインタレース映像出力ステップと、上記インタレースフレームのコマ落ちを検出するコマ落ち検出ステップと、上記コマ落ち検出ステップがコマ落ちを検出すると、コマ落ちの直前あるいは直後の上記インタレースフレームを構成するn枚のフィールドのうち、時間的にコマ落ち部に近い一枚のフィールドをn倍ライン補間することでプログレッシブフレームに変換し、コマ落ち部の画像を該画像で補うコマ落ち対処ステップと、上記インタレース映像出力ステップ及びコマ落ち対処ステップとが出力したフレームを、インタレースフレームとして、画像処理を行う画像処理ステップとを備えたことを特徴とすることで、時間的成分の欠落をコマ落ち部分のみにとどめ、請求項９で述べた効果を実現できる。

【図面の簡単な説明】

【図１】本発明の第１の実施の形態の映像情報処理装置の構成図

【図２】本発明の第１の実施の形態の映像情報処理装置の動作を説明する図

【図３】プログレッシブ変換を説明する図

【図４】本発明の第１の実施の形態の映像情報処理装置で符号化した映像符号化情報を復号化した時の動作を説明する図

【図５】本発明の第２の実施の形態の映像情報処理装置の構成図

【図６】本発明の第２の実施の形態の映像情報処理装置の動作を説明する図

【図７】本発明の第２の実施の形態の映像情報処理装置で符号化した映像符号化情報を復号化した時の動作を説明する図

【図８】本発明の第３の実施の形態の映像情報処理装置の構成図

【図９】本発明の第３の実施の形態の映像情報処理装置の動作を説明する図

【図１０】本発明の第３の実施の形態の映像情報処理装置で符号化した映像符号化情報を復号化した時の動作を説明する図

【図１１】インタレースフレームを説明する図

【図１２】従来技術による映像情報処理装置の構成図

【図１３】従来技術による映像情報処理装置の動作を説明する図

【図１４】従来技術による映像情報処理装置で符号化した映像符号化情報を復号化した時の動作を説明する図

19　　20

【図１５】従来技術による映像情報処理装置においてコマ落ちが発生した時の動作を説明する図

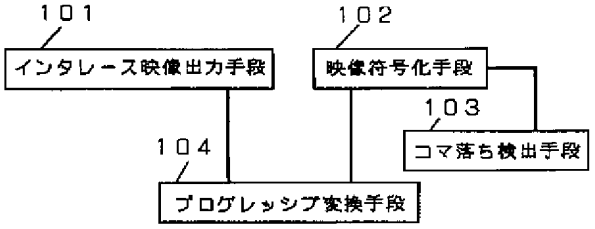
【図１６】従来技術による映像情報処理装置で符号化した映像符号化情報を復号化した際の不具合を説明する第１の図

【図１７】従来技術による映像情報処理装置で符号化した映像符号化情報を復号化した際の不具合を説明する第２の図

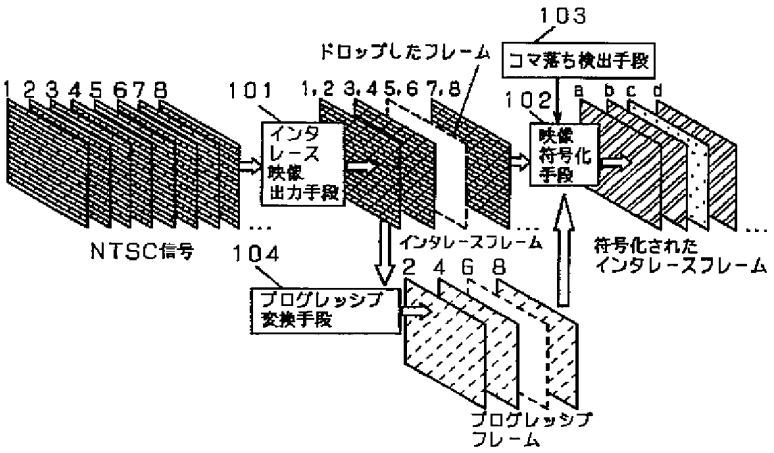
【符号の説明】

１０１　インタレース映像出力手段
１０２　映像符号化手段
１０３　コマ落ち検出手段
１０４　プログレッシブ変換手段

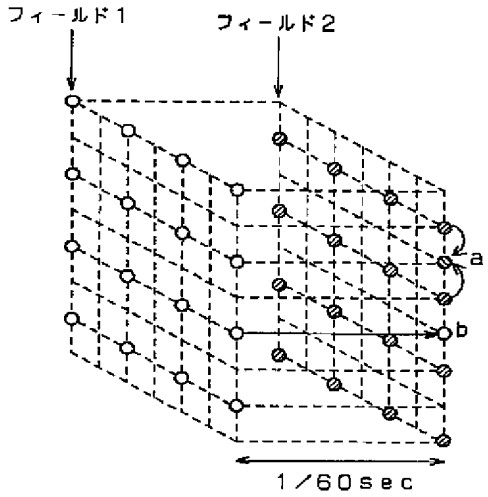
【図１】



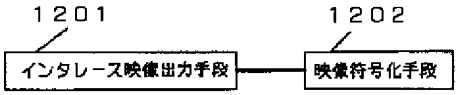
【図２】



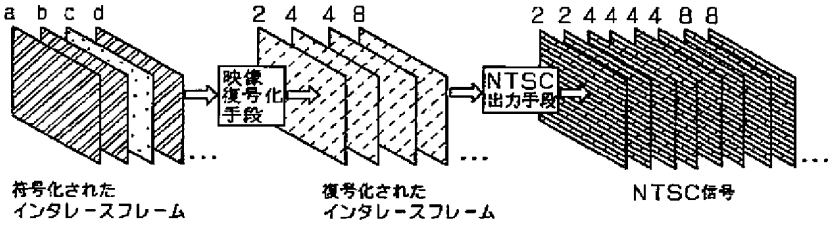
【図３】



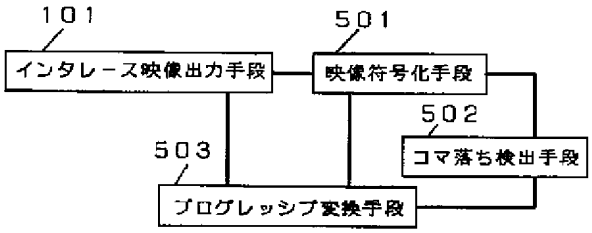
【図１２】



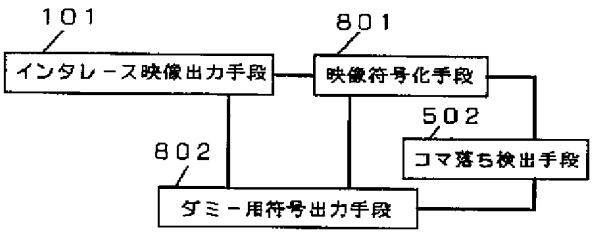
【図４】



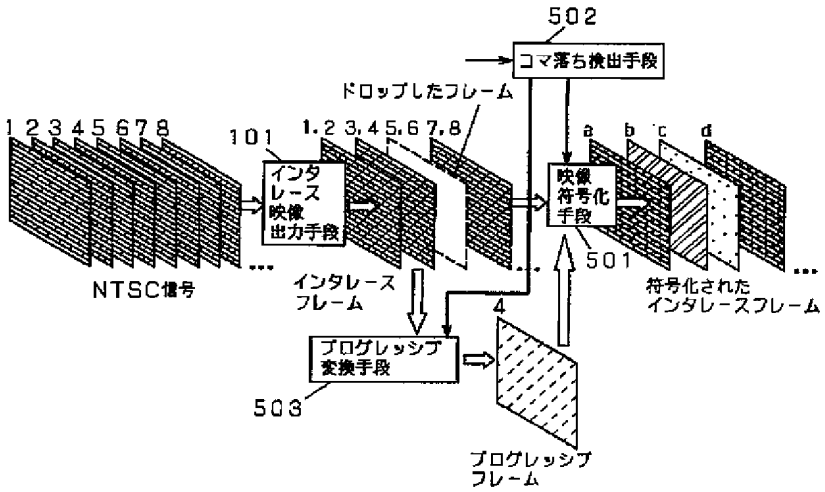
【図５】



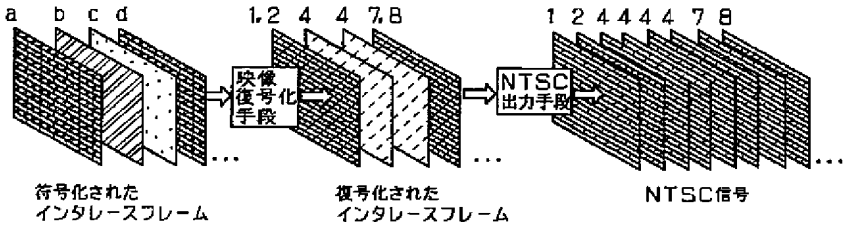
【図８】



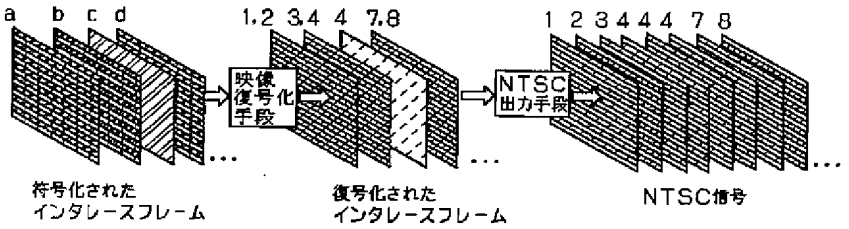
【図６】



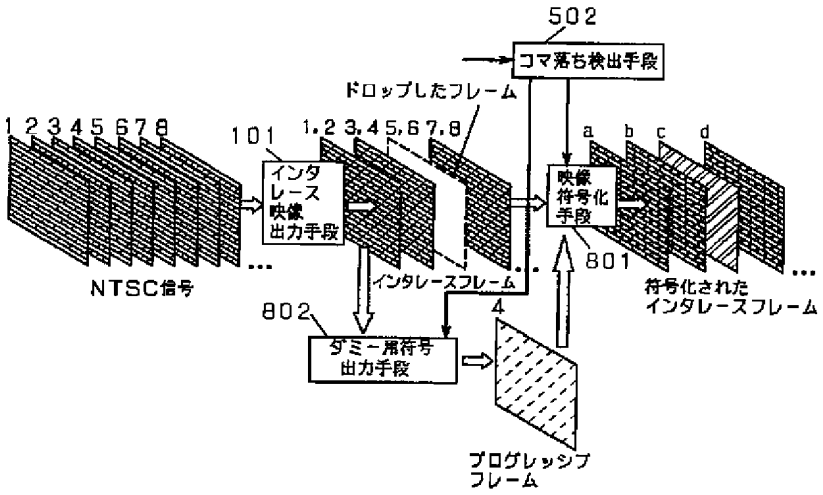
【図７】



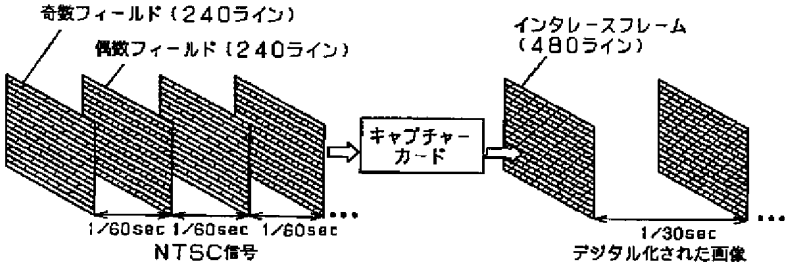
【図１０】



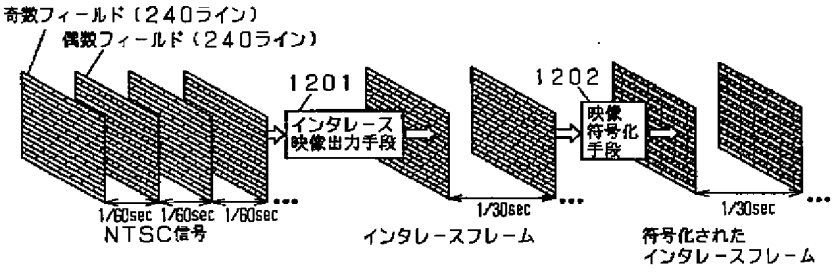
【図９】


502
コマ落ち検出手段
ドロップしたフレーム
1 2 3 4 5 6 7 8
101
インタレース映像出力手段
1 2 3 4 5 6 7 8
a b c d
映像符号化手段
801
NTSC信号
インタレースフレーム
符号化されたインタレースフレーム
802
ダミー用符号出力手段
4
プログレッシブフレーム

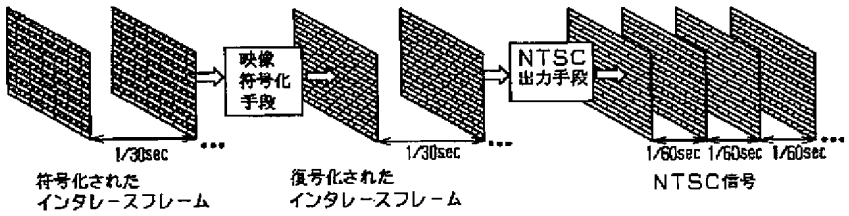
【図１１】


奇数フィールド（240ライン）
偶数フィールド（240ライン）
キャプチャーカード
インタレースフレーム（480ライン）
1/60sec 1/60sec 1/60sec
NTSC信号
1/30sec
デジタル化された画像

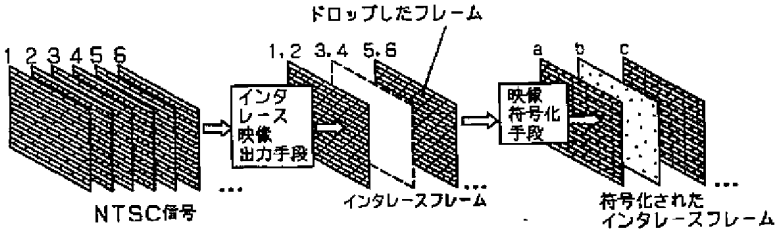
【図１３】


奇数フィールド（240ライン）
偶数フィールド（240ライン）
1201
インタレース映像出力手段
1202
映像符号化手段
1/60sec 1/60sec 1/60sec
NTSC信号
1/30sec
インタレースフレーム
1/30sec
符号化されたインタレースフレーム

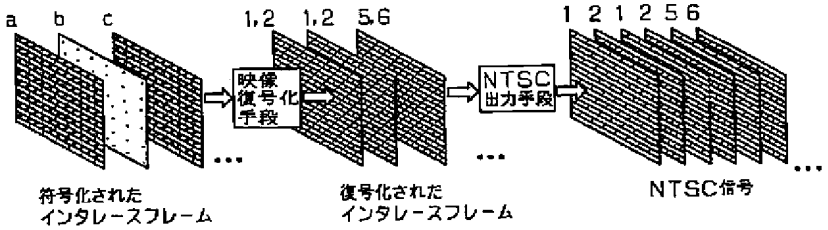
【図１４】



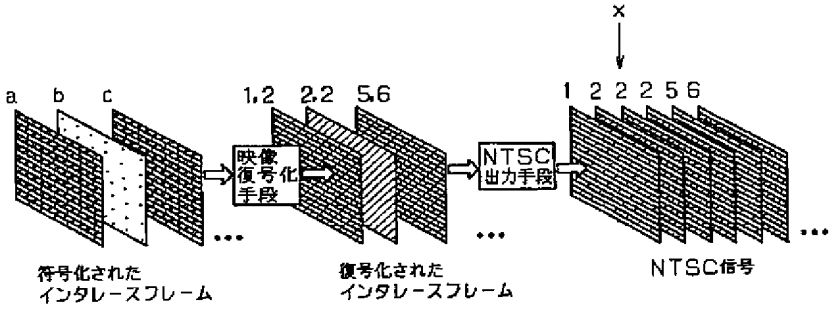
【図１５】



【図１６】



【図１７】



Reference Number: B065213
Mailing Number: 013879
Mailing Date: January 12, 2010

NOTICE OF REASONS FOR REJECTION

Patent Application No.:	2006-120296
Drafting Date:	January 6, 2010
Patent Office Examiner:	Akiyo Sakamoto 2954 5C00
Attorney for Patent Applicant:	Atsushi Aoki (and four others)
Applicable Provisions:	Article 29, paragraph 2; Article 36

It is deemed that the present application should be rejected for the following reasons. An argument, if any, should be submitted within 3 months from the mailing date of this notice.

REASONS

a. The inventions described in the following claims of this application are deemed ones which could easily have been made, prior to the filing of the patent application, by a person with ordinary skill in the art to which the inventions pertain, on the basis of the inventions described in the following publications distributed or the inventions made available to the public through electronic telecommunications lines in Japan or a foreign country prior to the filing of the patent application and, therefore, are unpatentable under the provisions of Article 29, paragraph 2, of the Patent Law.

NOTE (For the numbers of the cited references, see the List of Cited References)
Claims: 1 to 26
Reference: 1
Remarks:

Reference 1 describes the use of a copied frame in decoding, when the code of "the same as the previous frame" relating to a dropped frame is decoded (specifically, refer to paragraphs [0008] and [0009]).

Therefore, the technology for inserting a decoded image into a buffer memory, and the technology for adding a frame number are both known. Thus, it is not deemed that there is a remarkable difficulty in inserting a decoded frame including the copied frame in a buffer memory, and adding a number to the frame in the invention described in Reference 1.

b. The description of the claims of this application does not satisfy the requirements prescribed in Article 36, paragraph 6, item 2 of the Patent Law on the following points.

Notes

(1) The feature "configuring the buffer memory to enter image frames" is described in claim 1, but whether the description of "configuring" simply means that the image frames enter the buffer memory, or the term "configuring" means performing any process in addition to the above meaning.

The same is true for other claims having the same descriptions.

(2) The feature "A computer program comprising code to be executed by a computer, the code including, code for..., code for..., code for..., and code for...,." is described in claim 18. However, whether or not the above description for example, means "A computer program to execute a procedure for..., a procedure for..., a procedure for, a procedure for...., and a procedure for ...." is unclear (refer to Part VII, Chapter 1, 1.1.1(2)(b) of Examination Guidelines for Patent and Utility Model).

If an amendment is made, it should be noted that no new matter can be added and if asserting that the amendment is obvious based on the descriptions of the specification, the relevant technological ground should be specifically identified.

If another reason for rejection is found, a further notice of reasons for rejection will be issued.

List of Cited References

1. Japanese Unexamined Patent Publication (Kokai) No. 11-177987

Record of Results of Prior Art Search

·Searched Fields IPC
H04N 7/24 - 7/68, H04N 5/76, H04N 5/765, H04N 5/80 - 5/956

This Record of Results of Prior Art Search does not constitute a reason for rejection.

Electronic Acknowledgement Receipt

EFS ID:	6928863
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Steven R. Funk
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	08-FEB-2010
Filing Date:	25-JAN-2006
Time Stamp:	10:23:31
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-7-TPO004C1.pdf	55077 7a99c5da7eb5f83391d81552ce21b8c6b4738c63	no	2

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	1449-7-TPO004C1.pdf	55899 939370a2fc0530fce9943cd6922580aab0d1c8c0	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Foreign Reference	JP11177987.pdf	816946 40bb07babee962a57b146991203b22f872e3d209	no	15
Warnings:					
Information:					
4	NPL Documents	JPOA-06012010.pdf	64315 a52583a2ef438f7736f0d43bbc497bbf99fc17ba	no	3
Warnings:					
Information:					
		Total Files Size (in bytes):	992237		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,996	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

The Japanese reference was first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of the foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided. Also attached is a copy of the English translation of the Japanese Office Action.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Steven R. Funk/
Steven R. Funk
Reg. No. 37,830

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	6496980	12/17/2002	Tillman et al.			
	6510553	01/21/2003	Hazra			
	6639943	10/28/2003	Radha et al.			
	6614936	09/02/2003	Wu et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	EP0898427	2/24/1999	EP			NA	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	5/7/1996	Elloumi et al., "Issues in Multimedia Scaling: the MPEG Video Streams Case", Emerging Technologies and Applications in Communications, IEEE Computer Society, May 7, 1996.
	2/1/2002	Hannuksela, "Enhanced Concept of GOP", Joint Video Team of ISO-IES MPEG 7 itu-T VCEG, 2/1/2002.
	10/12/1997	Soderquist et al., "Memory Traffic and Data Cache Behavior of an MPEG-2 Software Decoder", IEEE Computer Society, October 12, 1997.
	2/2/2010	Copy of Office Action of related European application no. 06101013.8 dated 02/02/2010.
	2/2/2010	Copy of Office Action of related European application no. 06101014.6 dated 02/02/2010.
	3/16/2006	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 3/16/2006, 4 pages.
	8/10/2006	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 8/10/2006, 16 pages.
	9/11/2007	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 9/11/2007, 4 pages.
	3/13/2007	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 3/13/2007, 7 pages.
	12/7/2009	Copy of Notice of Allowance of parallel U.S. Application No. 10/306,942 dated 12/7/2009, 8 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

	5/6/2006	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 8/6/2006, 13 pages.
	11/9/2006	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 11/9/2006, 16 pages.
	5/31/2007	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 5/31/2007, 12 pages.
	7/9/2009	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 7/9/2009, 12 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

Issues in Multimedia scaling: the MPEG Video Streams Case

Omar Elloumi Hossam Afifi Laurent Toutain
Networks and Multimedia Services Department-Telecom Bretagne

Abstract:

Multimedia streams usually have an inherent hierarchy and an inter-inpendency of subsequent blocks in their data structures. Dealing with these features within the operating system and the network brings a global resource usage improvement. We propose in this paper a new method for media scaling based on two complementary techniques: Circular Buffers and Filtering. We describe an MPEG distributed application architecture built with the two combined techniques and show improvements in the received video quality.

1. Introduction

Scaling[1] [2],[3],[4],[10],[11] is a dynamic adjustment of traffic load by selective elimination in response to network overload or data buffers overflow. This is accomplished in end user system queues or in intermediate network nodes. Heterogeneity and multimedia multicast communications introduce bottlenecks and are major reasons for down scaling. We present a scaling mechanism that enables a weighted data flows multicast in heterogeneous environments. It fits into each capability and down-scales data streams only if this is really necessary. It can be implemented in intermediate and end-system nodes in a heterogeneous network in the presence of path bottlenecks. This paper focuses on scaling mechanism for end-system nodes[2]. We also describe its integration in the System V Streams[3] modules.

This paper is organized as follows: section 2. presents the MPEG video stream organization. Section 3. describes protocol stacks in UNIX systems and the Streams concept. The pilot application and the new protocol design are described in sections 4. and 5. Section 6. and 7. comment the performance results that we obtained and show the improvement when using the proposed scaling mechanism.

2. MPEG video streams properties

MPEG [1] streams are composed of sequences of coded frames grouped into GoPs (Group of Pictures). The video coding model uses three frame types: I (Intra-coded) frames are self-contained coded information; P (Predictive-coded) frames are the result of a motion compensated prediction, they depend upon the past I or P-picture; B (Bidirectional predictive-coded) frames are coded using motion compensated from a past and a future I or P-picture. A GoP video sequence can be decoded independently[4].

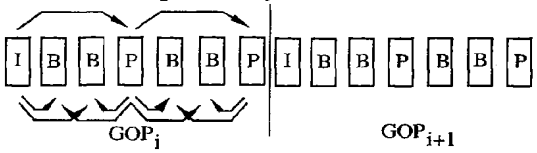


Figure 1 Frame dependency within one GOP

Figure 1 shows GoP internal dependencies. One can see that dropping any of the B pictures will not affect the GoP semantics. In contrast dropping a P picture leads to the drop of all dependent B or P pictures. This organization leads to many sub-GoP structures combination. Dropping one complete GoP do not affect the video stream structure. This induces two MPEG video features that will be used to design two scaling techniques[3]: 1) *Intra-GoP Scaling*: frame dropping within one GoP and *2) Inter-GoP scaling:* dropping complete GoPs eventually affected by *Intra-GoP scaling*.

Although a priority for each frame can be used (resulting on multiple Intra-GoP scaling possibilities), we attribute (to simplify the study) priority 1 to I frames, 2 to P frames and 3 to B ones.

3. Protocol stack design in UNIX systems

3.1. Queuing

Conventional Queuing in Unix hosts follows a simple FIFO policy [5]. New incoming data segments are stored in an input queue. When the queue is full, last incoming segments are dropped (also called Drop Tail technique) until an application consumes seg-

1. Also called *"down scaling"*
2. Hierarchical distributed scaling is studied in [8]
3. we note by Stream with capital S the protocol design architecture in the system V terminology
4. Except for Open GoPs.

0-8186-7585-3/96 $5.00 © 1996 IEEE

3NSDOCID: <XP_____10164870A__I_>

ments from the queue head, releasing memory for future data arrival.

Such a queuing model is not suited to real time multimedia environments for two reasons. First, in order to keep an acceptable visual quality when dealing with video applications, it is better to drop older video data segments rather than newer incoming ones. And second, the Drop Tail policy can induce an incoherence in the video stream segmented on multiple datagrams (applications usually flush the buffer and wait for a synchronization point). We propose an application requirement oriented scalable dynamic mechanism. It emulates GoPs batch arrivals in order to be able to drop complete semantically correlated set of datagrams.

3.2. Data copying

Data copying introduces penalizing delays. A first copy happens from the network interface private memory to the kernel's memory. A second copy into the application space is necessary. Applications that are penalized with losses in the network or operating system are usually forced to drop additional data segments. This is preferably achieved before copying data from the kernel. Implementing scaling procedure before data is copied can thus remove unnecessary data that cannot be used by applications. This can be done in UNIX system V which proposes the Streams facility.

A Stream [7] is composed of a head of Stream, a set of modules and/or drivers and a device driver (network interface driver), all of them organized in a vertical stack. Data is hence processed in pipeline where each module introduces its treatment and handles data through pointers (logical copy) to the next module. This architecture advantage is that no copies are performed until data reaches the head of Stream for incoming data or the device driver for outgoing data.

4. Protocol stack architecture

This section describes an architecture intended to deal with *Inter-GoP* and *Intra-GoP* scaling. *Intra-GoP* is implemented through MPEG stream de-multiplexing on multiple weighted logical flows and a filtering mechanism at the receiver. *Inter-GoP* scaling is achieved through Circular Buffers (CB)[3].

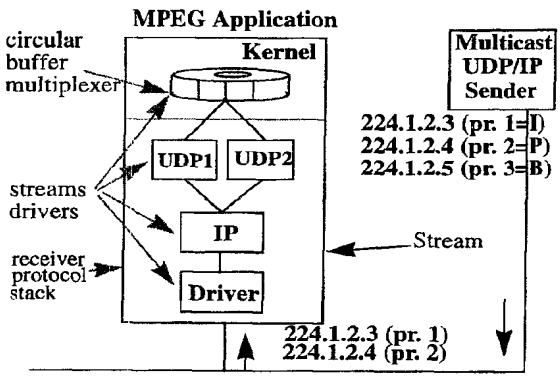


Figure 2 Streams Circular Buffer module for System V5

These two scaling techniques are described hereafter.

We have designed a priority based frame dropping mechanism using Circular Buffers. This mechanism was implemented for two different network protocols. In the first trial we used PTP Protocol [9] which offers a special field in the header for data priority. With the help of priorities we can apply filters in order to discard some logical flows. In the second trial, in order to implement a zero copy mechanism for the scaled data, we used different IP (Internet Protocol) Class D multicast addresses to differentiate data priorities. On Figure 2 priority 1 (I frames) is associated to class D 224.1.2.3 address. Application decides to filter, at the interface level, incoming flows by using the appropriate multicast addresses. Two fields are added, in order to be used by the CB module, to the datagrams (in the data part): Sequence Number and Information Unit (IU) number. The first is used to resolve desequencing that can be introduced either by the network, or by the scheduling of the UDP modules. The Information Unit (IU) number is used to emulate batch arrivals and hence to drop complete oldest GoPs[1] whenever the consumer application is slower than the arriving packets process. The whole implementation is inserted as a pseudo-driver multiplexer linked to the Stream (UDP protocol) (Figure 2).

The mechanism can easily be used for down-scaling on hosts acting as routers [8]. We only need to specify a priority for multi-

1. All datagrams within a GoP have the same IU number

BNSDOCID: <XP_____10164870A__I_>

casted packets. The multimedia router process will act as a filter and will transmit a reduced set of the incoming packets.

5. Application Design

Our application uses common Transport Layer Interface (TLI) of the Unix System V, to send datagrams over the previously mentioned priority class D UDP Streams. Each GoP is segmented on packets that have the size of the network Maximum Transfer Unit (MTU) minus protocols overhead. Three bytes are reserved at the beginning of each packet: one byte is used to indicate the Sequence Number and the two other bytes for IU.

Each Packet is sent on the correspondent Stream based on its priority.

The receiving application opens the CB Multiplexer and instantiates the multicast addresses corresponding to the flows it wants to receive. The multiplexer is linked to the UDP Stream. Coherent GoP structure is delivered to the CB when using increasing Class D addresses (and hence priorities).

6. Performance of a MPEG Receiver

A public domain MPEG player has been modified to deal with multicast facility.

The application instantiates UDP/IP Streams with the highest affordable priority (corresponding to multicast addresses) specified by the user. The interface discards all logical flows that have a higher priority (*Intra-GoP scaling*). In case of CPU bottleneck, the CB resolves overflow problems by overwriting, at the incoming data rate, complete oldest GoPs. This represents *Inter-GoP scaling* of IUs (corresponding each to a GoP eventually affected by the filtering: *Intra-GoP scaling*).

We concluded that priority improves video streams quality on loaded hosts. We have noticed that displaying the complete GoP on a loaded computer introduces important subsequent losses of complete GoPs. Using however a lower priority improves the regularity of the video stream. This effect was measured and plotted. Three operation zones are distinguished in the next section.

7. Performance measurements

This section studies how to improve the visual quality of the received stream when varying the CPU load. This is equivalent to varying the sending rate (incoming GoPs rate).

Scaling performance of the circular buffers module is shown in Figure 3 and Figure 4.

Figure 3 shows the percentage of GoP lost on a Sun Sparc Station 5 running Solaris 2.4 with circular buffer module inserted as mentioned in the UDP/IP Stream. Ethernet conventional interface was used in order to implement the multicast facility. The CIF video sequence has I:P:B ratio of 1:2:2 with a GoP of seven images (Figure 1).

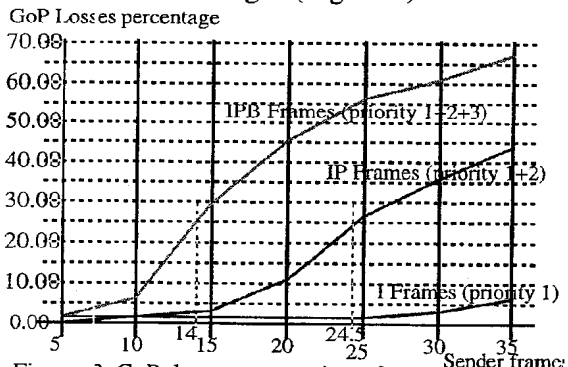


Figure 3 GoP losses at receiver for three increasing priorities

The horizontal axis shows the sender frame rate generation. The vertical axis gives the percentage of GoPs lost. Three curves are plotted with the three previously mentioned priorities. A high GoP loss percentage corresponds to a long visual irregularity in the video sequence. The first curve (priority 1) shows a very low percentage of lost GoPs. The second curve (priority 1+2) gives a larger number of subsequent decoded frames but also a larger number of losses. The third curve (priority 1+2+3) gives the highest number of GoP losses. We can conclude from this figure that when a loss happens in the network or in the operating system it will affect more seriously streams that have an IPB organization than streams that are less correlated like streams built on I frames only. Giving a lower priority for the latest stream can hence improve the global performance.

Figure 4 gives the ratio of the total captured and displayed frames for the different priorities.

BNSDOCID: <XP_____10164870A__I_>

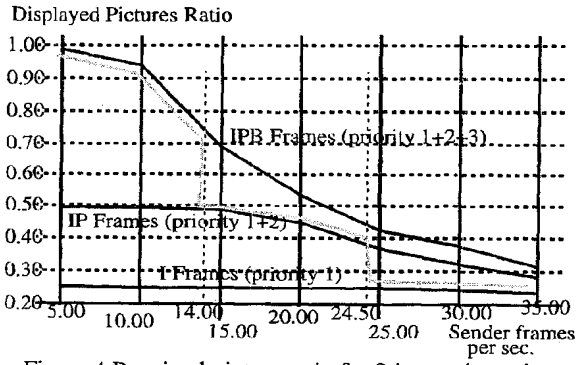


Figure 4 Received picture ratio for 3 increasing priority applications

It is clear from Figure 4 that the "IPB priority" decodes globally the largest number of frames. It is shown however that it suffers from large GoP losses when increasing the frame rate (interval 14->). It is hence visually better to choose a lower "IP priority" resulting in a smaller amount of decoded frames but with a greater visual regularity of pictures. This is depicted by the resulting gray broken curve in Figure 4. A simple model can hence be implemented in the application to receive the adequate number of priority streams. We can combine the two figures to implement a selective priority mechanism. From Figure 3, we can set arbitrarily a certain GoP loss threshold (e.g. 25%). This threshold divides the plots into three zones (at 14 and 24.5 frames per second). We then map these values on Figure 4. This also divides the plots into three working zones. The resulting mechanism will work with priority 1+2+3 up to 14 frames per second. When the decoder receives a larger amount of frames it will work with priority 1+2 and so on. It is to be mentioned that the chosen threshold can be studied taking into account psycho-visual considerations.

8. Conclusions

We have presented a scaling mechanism especially adapted for MPEG video coding. Two complementary methods are implemented in the system V Streams in order to prevent expensive kernel copies and to globally improve visual quality. The proposed scaling algorithm is used by MPEG receiver in a multicast environment. We show how scaling can be adapted to offer local QoS fitting at best with the CPU load. The scaling module can be used for all multimedia streams supporting hierarchical coding.

9. References

[1] CCITT SG XV Working Party XV/1. *Experts group on ATM Video Coding. Coded Representation of Picture and Audio Information.* MPEG 92/N0245.

[2] L. Delgrossi et Al. Media *Scaling for AudioVisual Communications with Heidelberg Transport System.* Proceedings of the ACM Multimedia 1993. San Diego. CA.

[3] O. Elloumi. *Dealing with Priorities in the PTP protocol.* Master Thesis. RSM. Telecom Bretagne. January 1995.

[4] S. Gumbrich, B. Kownatzki, J. Sandvoss. *Media Scaling for Real-Time Packet Video Transport.* Proceedings of the 4th Open Workshop on High Speed Networks. Brest, France. September 1994.

[5] Leffler S. McKusick. Karels M. Quarterman J. *The Design and Implementation of the 4.3BSD Unix Operating System.* AddisonWesley. 1989.

[6] C. Papadopoulos, G. Parulkar. "*Experimental Evaluation of SunOS IPC and TCP/IP protocol Implementation*". IEEE/ACM Trans. on Networking, Vol.1 No.2 April 1993.

[7] SunSoft Solaris 2.4 *Streams Programmer's Guide*

[8] F. Toutain, L. Toutain. "*Network Support for Multimedia Communications Using Distributed Media Scaling.*" Proceedings of COST 237 Workshop (Copenhagen), november 1995.

[9] L. Toutain, H. Afifi, J.P. LeNarzul. "*The Prism Distributed Multimedia Platform*". Proceeding of the IEEE Symposium in Advances in Computer Communications. Alexandria June 1995.

[10] H. Wittig, J. Winckler, J. Sandvoss "*Network Layer Scaling: Congestion Control in Multimedia Communication with heterogeneous Networks and Receivers.*" LNCS 882. Springer Verlag 1994.

[11] N. Yeadon. *Supporting Quality of Service in Multimedia Communications Vias the Use of Filters.* Technical Report. MPG Research Group. Lancaster University. December 1993.

BNSDOCID: <XP_____10164870A__I_>

Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG
(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)
2nd Meeting: Geneva, CH, Jan. 29 - Feb. 1, 2002

Document: JVT-B042r1
Filename: JVT-B042r1.doc
Generated: 2002-01-23
Revised: 2002-02-06

Title: **Enhanced Concept of GOP**

Status: Document accepted by JVT

Purpose: Proposal

Author: Miska M. Hannuksela
Nokia Mobile Phones
P.O. Box 68
33721 Tampere
Finland

Tel: +358 40 5212845
Email: miska.hannuksela@nokia.com

Source: Nokia Corporation

---

1. INTRODUCTION

1.1 Overview of the Paper

The paper is organized as follows. In the introductory part of the document, techniques utilizing scalable video coding or multiple representations of the same video content are presented in section 1.2 and a brief review to temporal scalability in video coding is given in section 1.3. The paper proposes means to signal temporal scalability based on chains of inter-dependent frames. An independently disposable chain of pictures is called a sub-sequence. The proposal is presented in section 2, and some examples of scalably coded bitstreams are given in section 3. After that, the document includes the patent disclosure form and some references.

The r1 revision of the document contains the proposal in the form that it was approved in the second JVT meeting. The changes compared to the original revision can be summarized as follows:

- The text for normative decoder operation was clarified and the terminology was fixed.
- The proposal does not require changes in transmitted packet or bit stream. A remark of that fact was added.
- The IPR statement was updated (a liberalizing comment on statement 2.2 was added).

1.2 Needs for Scalable Video Coding

1.2.1 Unequal Error Protection

Video streaming applications can often make use of unequal error protection (UEP). In order to apply UEP, video bitstreams have to be organized in portions of different importance in terms of visual quality. Techniques achieving this goal include data partitioning and scalable video coding. Examples of applicable UEP techniques are:

- Application-layer selective retransmission. Some commercial Internet streaming systems implement retransmission requests using proprietary protocols [1]. Work is going on in IETF to standardize a selective retransmission request mechanism as a part of RTP/RTCP [2][3].

BNSDOCID: <XP_____30005042A__I_>

A common feature for all of these retransmission request protocols is that they are not suitable for multicasting to a large number of players, as the network traffic may increase drastically. Consequently, multicast streaming applications cannot rely on interactive packet loss control.

- Transport-layer forward error control. Transmitters can add such redundancy to the transmitted packet stream that helps receivers to recover possibly lost data. Examples of such techniques are use of forward error correction (FEC) packets according to RFC 2733 [4] and transmission of redundant copies of selected packets.

- Guaranteed network Quality of Service (QoS). A stream might be transmitted through a number logical channels, each of which may have a different guaranteed QoS. Examples of such systems are the QoS architecture of Universal Mobile Telecommunication System (UMTS) [5] and System for Audio/Visual Live Services and Applications (SALSA) [6]. In UMTS (release 4 and later), an application can request for a certain QoS class (conversational, streaming, interactive, or background) and specific values for QoS parameters within the requested QoS class, such as guaranteed bit-rate and SDU error ratio. SALSA works on top of a CATV system. It provides a 100-kbps reliable channel and a 300-kbps best-effort channel per connection.

- Differentiated Services (DiffServ) [7][8]. In DiffServ, packets are classified and marked to receive a particular per-hop forwarding behavior on nodes along their path. While DiffServ is not available throughout the Internet, many private networks support it.

1.2.2 Rate Shaping

The objective of rate or traffic shaping is to match the bit-rate of a pre-compressed video bitstream to the target rate constraint. Many networks are not capable of providing channels with constant throughput. Thus, many streaming servers monitor the network conditions and adjust the transmitted bitstream accordingly. One way to control the bit-rate of the transmitted stream is to decide which enhancement layer to transmit. Another way is to switch between multiple representations of the same content each targeted for different bit-rate.

1.3 Temporally Scalable Video Coding

1.3.1 Individually Disposable Pictures

Conventionally, B pictures have not been used as prediction references. Consequently, they have provided a way to achieve temporal scalability. (In the JVT codec, B pictures can also be stored are referenced in other pictures.)

H.263 Annex U and JVT JM-1 allow signaling that a particular picture need not be stored in the buffer of reference pictures. Consequently, such a picture can be safely disposed.

1.3.2 Disposal of Picture Chains

A known method in today's streaming systems to cope with drastically dropped channel throughput is to transmit INTRA pictures only. When the network throughput is restored, INTER pictures can be transmitted again from the beginning of the next Group of Pictures (GOP).

Generally, any chain of INTER pictures can be safely disposed, if no other picture is predicted from them. This fact can be utilized to treat INTER pictures at the end of a prediction chains less important than other INTER pictures. In some cases, such INTER pictures represent contents of scene transitions, such as fades, and can be disposed without losing essential pictorial information.

3NSDOCID: <XP_____30005042A__I_>

Reference picture selection enables different kinds of temporal scalability schemes, which are herein referred to as layered temporal scalability. One example of such a scheme is proposed in [6]. It is called SALSA and depicted in Figure 1. The "base" layer has a relatively low temporal resolution enabling it to be transmitted in a 100-kbps reliable channel. For example, every 6th coded frame may belong to the "base" layer. The "enhancement" layer increases the temporal resolution and is transmitted in a 300-kbps best-effort channel.

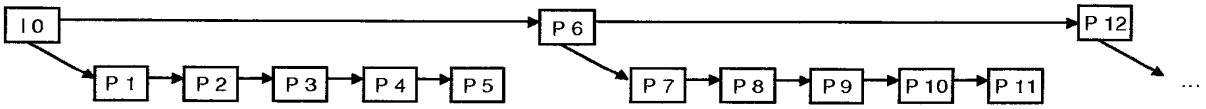


Figure 1. Temporal scalability proposed in SALSA.

A straightforward modification of the SALSA coding scheme targeted for rate shaping is illustrated in Figure 2. "SP 6" (and "SP 12") boxes indicate identical representations of frame 6 (and 12 respectively) based on different reference frames. A streaming server decides which coded SP frame is transmitted based on the temporal resolution of the previously transmitted section of the bitstream. The encoder may also switch to a lower or higher temporal resolution after such an SP frame. The compression efficiency of this kind of a coding scheme is nearly as good as in a non-scalable coding scheme as shown in VCEG-M73.

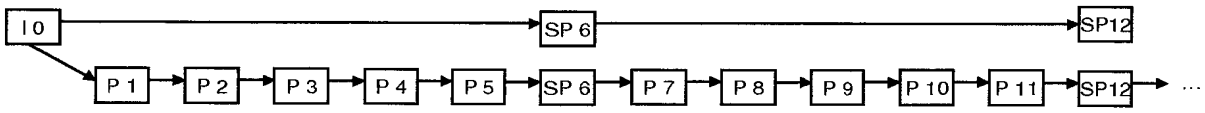


Figure 2. Temporal scalability for rate shaping.

In the JVT design, B pictures can be used as prediction references. Another example of temporal scalability scheme is illustrated in Figure 3, which presents an IBBP coding pattern, where the latter B picture is predicted from the former one. Pictures B3 and B6 can be disposed individually, whereas disposal of pictures B2 and B5 requires disposal of pictures B3 and B6 respectively.

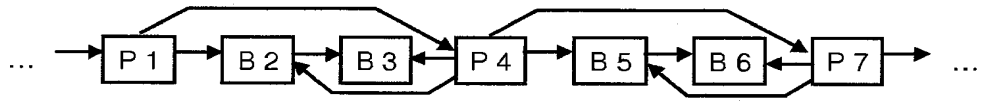


Figure 3. B pictures as motion compensation references in an IBBP coding pattern.

2. PROPOSAL

2.1 Overview

The proposal enhances the concept of independently decodable GOPs presented in JVT-B041 so that disposable chains of pictures can be easily identified and disposed. The proposal organizes pictures in temporal scalability layers and sub-sequences. A sub-sequence represents a number of inter-dependent pictures that can be disposed without affecting the decoding of any sub-sequence in any less enhanced scalability layer.

The result of sub-sequence or layer disposal must be a valid bitstream. Moreover, as the foreseen applications, namely unequal error protection and rate shaping, take place in a streaming server, no changes to Video Coding Layer syntax or any transport packetization syntax were seen necessary. Thus, we propose syntax changes only to the interim file format (VCEG-O58). The changes allow signaling of layers, sub-sequences and their dependencies. Appendix A of the document contains an updated version of the interim file format. The use of the proposed syntax changes is optional, i.e., non-scalable bitstreams can be created without signaling overhead. Furthermore, file format parsers can ignore the information related to scalability layers and sub-sequences safely.

Only entire sub-sequences and non-stored pictures should be disposed. When disposing a sub-sequence, any sub-sequence depending on the disposed one should be discarded as well. Otherwise, a mismatch in the multi-picture buffer contents may occur between the encoder and the decoder, which results into incorrectly reconstructed pictures in the decoder. Encoder constraints on multi-picture buffering are listed in section 2.3.

Decoders shall be informed if it is possible to dispose sub-sequences intentionally. This is done with a new flag in the parameter set (defined in VCEG-O58 and VCEG-N72r1). When intentional disposal is allowed, decoders shall not infer an unintentional picture loss immediately when a missing picture number is detected but rather update the multi-picture buffer so that the default index order remains the same as in the encoder. Detailed decoder operations are presented in section 2.4.

No changes in the transmitted packet or bit stream syntax are proposed. In other words, no information about layers and sub-sequences is transmitted.

The proposal is targeted for all "streaming" and "storage" applications.

2.2 Definition of a Layer and a Sub-Sequence

Layers can be ordered hierarchically based on their dependency on each other. The base layer is independently decodable. The first enhancement layer depends on some of the data in the base layer. The second enhancement layer depends on some of the data in the first enhancement layer and in the base layer and so on. The subjective quality increases along with the number of decoded enhancement layers.

A sub-sequence shall not depend on any other sub-sequence in the same or in a more enhanced scalability layer. In other words, it shall only depend on one or more sub-sequences in one or more less enhanced scalability layers. A sub-sequence in the base layer can be decoded independently of any other sub-sequences and prior long-term reference pictures. Thus, the beginning of a base layer sub-sequence can be used as a random access position.

A sub-sequence covers a certain period of time within the sequence. Sub-sequences within a layer and in different layers can partly or entirely overlap. A picture shall reside in one layer and in one sub-sequence only.

Pictures sharing the same display time in the same layer are considered alternative representations of the same picture content, and any representation of the picture can be used. These so-called sync pictures shall reside in the same sub-sequence, but their reference pictures can reside in any sub-sequence. Indication of the reference sub-sequence helps streaming servers to decide which coded sync frame to transmit.

A layer number is assigned to layers. Number zero stands for the base layer. The first enhancement layer is associated with number one and each additional enhancement layer increments the number by one.

A sub-sequence identifier is assigned to sub-sequences. Consecutive sub-sequences (within the same scalability layer) should not have the same identifier, but no other numbering rule for sub-sequence identifiers is necessary.

2.3 Constraints in Multi-Picture Buffering

If you are unfamiliar with multi-picture buffering, please refer to VCEG-O10r1 or to section 7 of TML-9 (VCEG-N83d1) to get an insight.

Any entity, such as a streaming server, disposing parts of a bitstream must make sure that the decoder is able to obtain essentially the same state of the multi-picture buffer as the encoder at any instant of time. Otherwise, the default picture order might differ in the decoder from the one used while encoding, and incorrect reference pictures may be used. Thus, the following constraints in memory management operations of the multi-picture buffer have to be followed when coding a bitstream according to the proposal:

- Marking a picture as "Unused" shall be done in a stored base-layer picture only. Otherwise, if a sub-sequence containing the assignment to "Unused" were disposed, the default index order of pictures might differ in the encoder and in the decoder.

- Resetting the multi-picture buffer shall be done in a stored base-layer picture only. Otherwise, the reset indication might be disposed and the contents of the multi-picture buffer would differ.

- Changing the number of long-term pictures shall be done in a stored base-layer picture only, as this operation affects the default index order.

- Assigning a long-term picture index to a short-term picture can be done in any picture in the same sub-sequence where the picture resides or in any of the sub-sequences used for predicting the current one, if the long-term picture index was already taken into use in one of these sub-sequences. Otherwise, if the long-term picture index was "Unused" or was taken into use in a sub-sequence independent from the current one, a long-term index shall be assigned in the base layer. (Re-using a long-term index does not change the default index order, whereas taking a fresh long-term index into use does.)

2.4 Decoder Operation

The defined reaction to a gap in picture numbers (a paragraph in section 7.6.2 of VCEG-N83d1) is as follows:

"If the decoder detects a missing picture, it may invoke some concealment process, and may insert an error-concealed picture into the multi-picture buffer. Missing pictures can be identified if one or several picture numbers are missing or if a picture not stored in the multi-picture buffer

BNSDOCID: <XP_____30005042A__I_>

is indicated in a transmitted ADPN or LPIR. Concealment may be conducted by copying the closest temporally preceding picture that is available in the multi-picture buffer into the position of the missing picture. The temporal order of the short-term pictures in the multi-picture buffer can be inferred from their default relative index order and PN fields. In addition or instead, the decoder may send a forced INTRA update signal to the encoder by external means (for example, Recommendation H.245), or the decoder may use external means or back-channel messages (for example, Recommendation H.245) to indicate the loss of pictures to the encoder. A concealed picture may be inserted into the multi-picture buffer when using the "Sliding Window" buffering type. If a missing picture is detected when decoding a Slice, the concealment may be applied to the picture as if the missing picture had been detected at the picture layer."

As the suggested reaction is correct for accidentally lost pictures only, we propose that the possibility of intentional sub-sequence disposal is indicated in the parameter set. We call the indication as the required picture number update behavior indication. The value of the indication (on or off) in the referred parameter set shall be the same as in the previously transmitted NAL packets, unless a multi-picture buffer reset is associated with the current NAL packet.

If the required picture number update behavior is signaled, a gap in picture numbers shall be treated as follows:

- Decoders shall not infer an accidental loss immediately.
- Decoders shall not invoke any immediate concealment.
- Decoders shall generate the default index order as if pictures corresponding to the lost picture numbers were inserted to the multi-picture buffer using the "Sliding Window" buffering type. An index corresponding to a missing picture number is called an "invalid" index.

Decoders shall infer an accidental picture loss if any "invalid" index is referred to in motion compensation or if an "invalid" index is re-mapped. As a reaction to a detected picture loss, decoders may invoke some concealment process similarly to what has been defined in section 7.6.2 of VCEG-N83d1. If the subject of a memory management control operation is a picture corresponding to an "invalid" index, the operation shall be carried out normally.

We propose that the paragraph in section 7.6.2 of VCEG-N83d1 is replaced with the following paragraph

"If the indicated parameter set in the latest received slice or data partition signals the required picture number update behavior, the decoder shall operate as follows. The default picture index order shall be updated as if pictures corresponding to missing picture numbers were inserted to the multi-picture buffer using the "Sliding Window" buffering type. An index corresponding to a missing picture number is called an "invalid" index. The decoder should infer an unintentional picture loss if any "invalid" index is referred to in motion compensation or if an "invalid" index is re-mapped.

If the indicated parameter set in the latest received slice or data partition does not signal the required picture number update behavior, the decoder should infer an unintentional picture loss if one or several picture numbers are missing or if a picture not stored in the multi-picture buffer is indicated in a transmitted ADPN or LPIR.

In case of an unintentional picture loss, the decoder may invoke some concealment process. If the required picture number update behavior was indicated, the decoder may replace the picture corresponding to an "invalid" index with an error-concealed one and remove the "invalid" indication. If the required picture number update behavior was not indicated, the decoder may insert an error-concealed picture into the multi-picture buffer assuming the "Sliding Window"

buffering type. Concealment may be conducted by copying the closest temporally preceding picture that is available in the multi-picture buffer into the position of the missing picture. The temporal order of the short-term pictures in the multi-picture buffer can be inferred from their default relative index order and PN fields. In addition or instead, the decoder may send a forced INTRA update signal to the encoder by external means (for example, Recommendation H.245), or the decoder may use external means or back-channel messages (for example, Recommendation H.245) to indicate the loss of pictures to the encoder."

2.5 Sub-Sequences Compared to Alternate Tracks

The interim file format (VCEG-O58) contains a mechanism to include multiple coded representations of the same uncompressed video sequence into the same file. These representations are referred to as alternate tracks. The interim file format also allows signaling of pictures that can be used to switch between alternate tracks.

Most (if not all) patterns of coded pictures, such as the one illustrated in Figure 2 can be expressed both using alternate tracks and using temporal scalability layers and sub-sequences. We propose that the alternate track mechanism would be used for rough bit-rate adjustment, as it does not provide means for indicating scalability layers, dependencies between pictures, and information about temporary bit- or frame rates. The sub-sequence mechanism can be considered more flexible in bit-rate scalability. It suits better for short-term, immediate, and accurate bit-rate adjustment.

3. EXAMPLES

The notation used in the figures is as follows: I stands for INTRA picture and P for INTER pictures. Picture type is followed by two digits separated by a dot (x.y). x stands for sub-sequence identifier and y stands for picture number. Relative capturing and displaying time runs from left to right. Scalability layers are on top of each other and indicated on the left.

Figure 4 shows an example of a scene transition, where the first scene is faded to black. The faded pictures are located in an enhancement layer, as they are not considered to contain any subjectively essential information. The former base layer sub-sequence happens to have identifier 0 and the latter one has identifier 1, whereas the sub-sequence in the enhancement layer happens to have identifier 3. Picture numbers run independently from sub-sequences but are reset to zero at the beginning of each base-layer sub-sequence as proposed in JVT-B041.

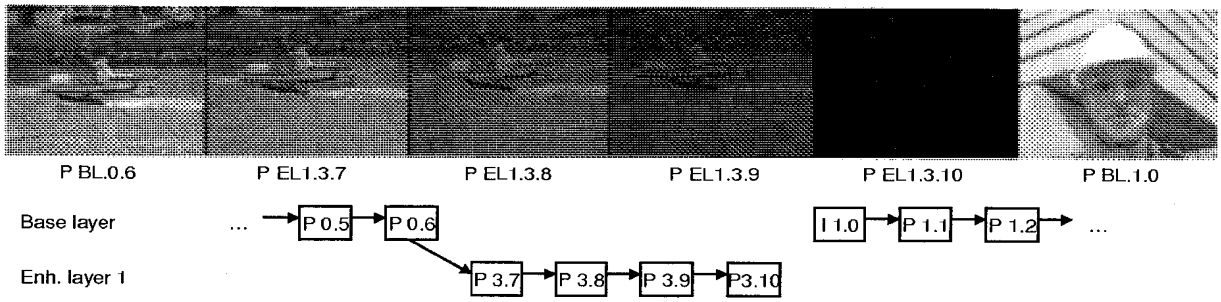


Figure 4. Fade to black.

Figure 5 shows how the temporal scalability scheme proposed in SALSA can be handled with layers and sub-sequences. Notice that even though the sub-sequence in the base layer and the first sub-sequence in the enhancement layer are given the same sub-sequence identifier, they can be distinguished from each other, since they reside in different layers.

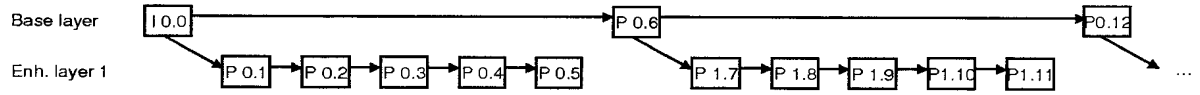


Figure 5. Temporal scalability proposed in SALSA.

Figure 6 shows an example of sub-sequence and picture numbering for a temporal scalability scheme for rate shaping purposes. Sub-sequence and picture numbering is similar to the previous example.

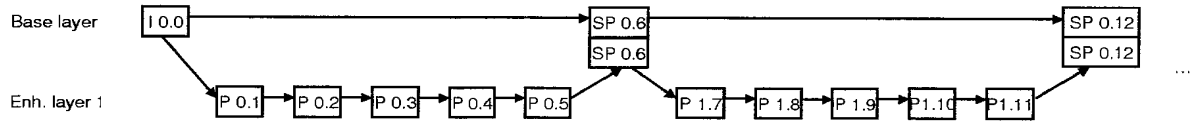


Figure 6. Temporal scalability for rate shaping.

Figure 7 shows an example of sub-sequence and picture numbering, when some B pictures are used as motion compensation reference pictures. Picture number 7 is assigned to both picture B EL1.1.7 and picture P BL.0.7, as the B picture is not stored in the multi-picture buffer. However, as picture B EL1.1.6 is stored in the multi-picture buffer, the picture number is incremented normally in the picture following it.

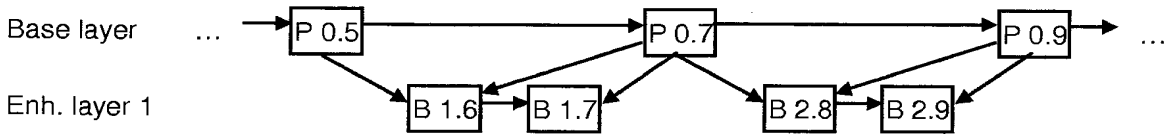


Figure 7. B pictures as motion compensation references in an IBBP coding pattern.

3NSDOCID: <XP_____30005042A__I_>

4. IPR STATEMENT

International Telecommunication Union
Telecommunication Standardization Sector

International Organization for Standardization

International Electrotechnical Commission

Joint Video Coding Experts Group - *Patent Disclosure Form*

(Typically one per contribution and one per Standard | Recommendation)

Please send to:

JVT Rapporteur Gary Sullivan, Microsoft Corp., One Microsoft Way, Bldg. 9, Redmond WA 98052-6399, USA
Email (preferred): Gary.Sullivan@itu.int Fax: +1 425 706 7329 (+1 425 70MSFAX)

This form provides the ITU-T | ISO/IEC Joint Video Coding Experts Group (JVT) with information about the patent status of techniques used in or proposed for incorporation in a Recommendation | Standard. JVT requires that all technical contributions be accompanied with this form. *Anyone* with knowledge of any patent affecting the use of JVT work, of their own or of any other entity ("third parties"), is strongly encouraged to submit this form as well.

This information will be maintained in a "living list" by JVT during the progress of their work, on a best effort basis. If a given technical proposal is not incorporated in a Recommendation | Standard, the relevant patent information will be removed from the "living list". The intent is that the JVT experts should know in advance of any patent issues with particular proposals or techniques, so that these may be addressed well before final approval.

This is not a binding legal document; it is provided to JVT for information only, on a best effort, good faith basis. Please submit corrected or updated forms if your knowledge or situation changes.

This form is *not* a substitute for the *ITU ISO IEC Patent Statement and Licensing Declaration*, which should be submitted by Patent Holders to the ITU TSB Director and ISO Secretary General before final approval.

Submitting Organization or Person:	
Organization name	**Nokia Corporation**
Mailing address	
Country	
Contact person	
Telephone	
Fax	
Email	
Place and date of submission	Geneva, Switzerland, January 2002
Relevant Recommendation | Standard and, if applicable, Contribution:	
Name (ex: "JVT")	**JVT**
Title	**Enhanced Concept of GOP**
Contribution number	**JVT-B042**

(Form continues on next page)

Disclosure information – Submitting Organization/Person (choose one box)

☐ 2.0 The submitter is not aware of having any granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution.

or,

The submitter (Patent Holder) has granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution. In which case,

☐ 2.1 The Patent Holder is prepared to grant – on the basis of reciprocity for the above Recommendation | Standard – a free license to an unrestricted number of applicants on a worldwide, non-discriminatory basis to manufacture, use and/or sell implementations of the above Recommendation | Standard.

☒ 2.2 The Patent Holder is prepared to grant – on the basis of reciprocity for the above Recommendation | Standard – a license to an unrestricted number of applicants on a worldwide, non-discriminatory basis and on reasonable terms and conditions to manufacture, use and/ or sell implementations of the above Recommendation | Standard.

Such negotiations are left to the parties concerned and are performed outside the ITU | ISO/IEC.

☐ 2.2.1 The same as box 2.2 above, but in addition the Patent Holder is prepared to grant a "royalty-free" license to anyone on condition that all other patent holders do the same.

☐ 2.3 The Patent Holder is unwilling to grant licenses according to the provisions of either 2.1, 2.2, or 2.2.1 above. In this case, the following information must be provided as part of this declaration:

- patent registration/application number;
- an indication of which portions of the Recommendation | Standard are affected.
- a description of the patent claims covering the Recommendation | Standard;

*In the case of any box **other than 2.0** above, please provide the following:*

Patent number(s)/status _____

Inventor(s)/Assignee(s) _____

Relevance to JVT _____

Any other remarks: _____

(please provide attachments if more space is needed)

(form continues on next page)

Third party patent information – fill in based on your best knowledge of relevant patents granted, pending, or planned by other people or by organizations other than your own.

Disclosure information – Third Party Patents (choose one box)

[X] 3.1 The submitter is not aware of any granted, pending, or planned patents *held by third parties* associated with the technical content of the Recommendation | Standard or Contribution.

[ ] 3.2 The submitter believes third parties may have granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution.

For box 3.2, please provide as much information as is known (provide attachments if more space needed) - JVT will attempt to contact third parties to obtain more information:

3rd party name(s) _____

Mailing address _____
Country _____
Contact person _____
Telephone _____
Fax _____
Email _____
Patent number/status _____
Inventor/Assignee _____
Relevance to JVT _____

Any other comments or remarks:

To support a royalty-free baseline we follow ITU-T 2.2.1 for Baseline Profile applications agreeing to a free license on condition that all other patent holders do the same.

5. REFERENCES

[1] Gregory J. Conklin, Gary S. Greenbaum, Karl O. Lillevold, and Alan F. Lippman, "Video Coding for Streaming Media Delivery on the Internet", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 3, March 2001.

[2] A. Miyazaki, H. Fukushima, K. Hata, T. Wiebke, R. Hakenberg, C. Burmeister, N. Takatori, S. Okumura, T. Ohno, " RTP Retransmission Payload Format", IETF Internet Draft draft-ietf-avt-rtp-selret-02.txt, 18 July 2001.

[3] D. Leon, V. Varsa, " RTP retransmission framework", IETF Internet Draft draft-leon-rtp-retransmission-00.txt, July 2001.

[4] J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction" IETF RFC 2733, December 1999.

[5] 3GPP TS 23.107, "QoS Concept and Architecture (Release 4)", V4.2.0, October 2001, available via ftp://ftp.3gpp.org/Specs/2001-09/Rel-4/23_series/23107-420.zip.

[6] M. Reha Civanlar, "Video Over the Internet", invited speak, International Packet Video Workshop PV2001, Kyongju, South Korea, 1 May 2001, available via http://pv2001.kaist.ac.kr/.

[7] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", IETF RFC 2474, December 1998.

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services", IETF RFC 2475, December 1998.

Memory Traffic and Data Cache Behavior of an MPEG-2 Software Decoder

Peter Soderquist
School of Electrical Engineering
Cornell University
Ithaca, New York 14853
E-mail: pgs@cs.cornell.edu

Miriam Leeser
Dept. of Electrical and Computer Engineering
Northeastern University
Boston, Massachusetts 02115
E-mail: mel@ece.neu.edu

Abstract

We investigate the impact of multimedia applications on the cache behavior of desktop systems. Specifically, we consider the memory bandwidth and data cache challenges associated with MPEG-2 software decoding. Recent extensions to instruction set architectures, including Intel's MMX, address the computational aspects of MPEG decoding. The large amount of data traffic generated, however, has received little attention. Standard data caches consistently generate an excess of cache-memory traffic. Varying basic cache parameters only reduces traffic to double the minimum required at best. Incremental changes in cache size have a negligible effect for most feasible values. Increasing set associativity yields rapidly diminishing returns, and manipulating line size is similarly unproductive. Achieving higher efficiency requires understanding the composition and behavior of the decoder data set. We present a model of MPEG-2 decoder memory behavior and describe how to exploit this knowledge to minimize required memory bandwidth. Our results show that simply eliminating one component, video output data, from the cache can reduce traffic by as much as 50 percent.

1 INTRODUCTION

The capabilities of microprocessors are expanding to include multimedia functionality, supported by ISA extensions (e.g. Intel's MMX [7], Sun Microsystems' VIS [8] and Hewlett-Packard's MAX-2 [5]) and hardware enhancements. MPEG-2, the favored standard for video compression, is a popular target application for recent processor designs. These architectural modifications address the computational challenges of MPEG, yet the consequences for memory subsystems remain largely unexplored. The high data rate, large set sizes, and distinctive memory access patterns of MPEG exert a particular strain on caches. While miss rate levels are acceptable, standard caches, even very large ones, generate significant excess memory traffic.

We examine the traffic and data cache behavior of an MPEG-2 decoder running Main level streams on a general-purpose microprocessor. An analysis of the data types utilized by the decoder complements trace-driven cache simulations of representative streams. We show that there is a benefit to even small, simple caches. Unfortunately, manipulating standard parameters (cache size, associativity, and line size) fails to cost-effectively reduce excess memory traffic. These results suggest several architectural enhancements, which we outline. We also show that even relatively simple measures, such as selective caching of specific data types, can dramatically improve efficiency, reducing cache-memory traffic by up to 50%, and 25% on average, for a wide range of cache sizes.

The goal of this work is to establish how caches behave under the strain of software MPEG-2 decoding, and exploit that knowledge to achieve more efficient cache resource utilization. Successful application of these insights will raise the multimedia performance standard of microcomputers and support the deployment of numerous applications which are currently technologically infeasible. Facilitating the efficient utilization of resources can also enable the production of low-cost and even low-power portable systems, and a broader proliferation of digital video applications.

2 BACKGROUND

Consider the problem of decoding MPEG-2 in software on a general-purpose computing platform. The machine type of primary interest is a typical desktop PC or workstation, as envisioned in Figure 1. The diagram also illustrates the typical flow of video data in the system. Imagine the user is viewing, for example, an entry in a multimedia encyclopedia. Compressed data is transferred from the file system (e.g. CD-ROM or hard disk) by direct memory access (DMA) and buffered in the main memory. The CPU reads the data through its cache hierarchy and writes the decoded, dithered RGB video, one frame at a time, to main memory where it is stored in an image of the displayed video window. Finally, another DMA transfer brings the

1063-6404/97 $10.00 © 1997 IEEE

BNSDOCID: <XP_____10251768A__I_>

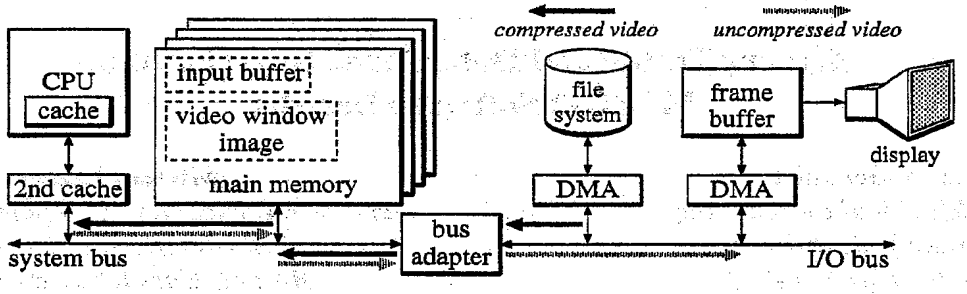


Figure 1: Typical desktop machine with software MPEG-2 decoding traffic

video data to the frame buffer where it eventually appears on the display.

It is clear that there is a large amount of data, much of it time-sensitive, being transferred through the system. Most of this is concentrated on the main system bus. At the very least, there are two streams each of encoded and decoded video being concurrently transferred in and out of main memory, amounting to a minimum sustained load of 63 Mbytes/s. This ignores the bandwidth required by other applications running on the system and sharing interconnect, main memory, and peripherals. Any excess memory traffic generated by cache inefficiency will further exacerbate this situation. Unfortunately, our simulations indicate that standard caches generate *at least* twice the minimum required level of cache-memory traffic, typically many times more, amounting to a significant strain on system capacity. The result is some combination of dropped frames, reduced frame rate, or degradation of overall system performance.

2.1 MPEG Overview

Appreciating the challenges of supporting video decoding requires some understanding of the MPEG standard [2]. MPEG attacks both the spatial and temporal redundancy of video signals to achieve compression. Video data is broken down into 8×8 pixel *Blocks* and passed through a discrete cosine transform (DCT). The resulting spatial frequency coefficients are quantized, run-length encoded, and then further compressed with an entropy coding algorithm. To exploit temporal redundancy, MPEG encoding uses motion compensation with three different types of frames. *I* (intra) frames contain a complete image, compressed for spatial redundancy only. *P* (predicted) MPEG frames are built from 16×16 fragments known as *macroblocks*. These consist primarily of pixels from the closest previous I or P frame (the *reference frame*), translated as a group from their location in the source. This information is stored as a vector representing the translation, and a DCT-encoded difference term, requiring far fewer bits then the original image fragment. *B* (bidirectional) frames can use the closest two I or P pictures – one before and one after in temporal order – as reference frames. Information not present in reference frames is encoded spatially on a block-by-block basis. All of data in P and B frames is also subject to run-length and entropy coding.

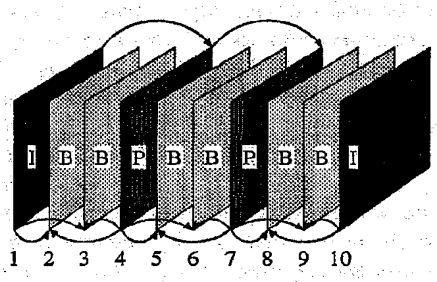


Figure 2: Typical sequence of MPEG frames, showing interframe dependencies

Figure 2 shows the interframe dependencies of the different frame types, superimposed on the displayed frame order. For decoding, these frames must be processed in the non-temporal order [1,4,2,3,7,5,6,10,8,9], which is a result of these dependencies. The dependencies between frames and the properties and and sequence of frame types determine in critical ways the flow pattern of MPEG data and the nature of hardware support required.

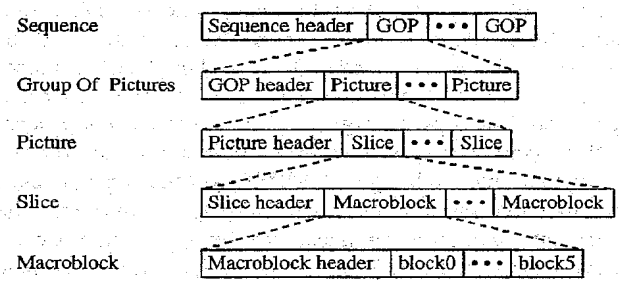


Figure 3: MPEG bitstream structure

The decoder reads the MPEG data as a stream of bits. Unique bit patterns, known as *startcodes* mark the division

BNSDOCID: <XP_____10251768A__I_>

between different sections of the data. The bitstream has a hierarchical structure, shown in simplified form in Figure 3. A *Sequence* (video clip) consists of *groups of pictures (GOP's)*. A GOP contains at least one I frame and typically a number P and B frames; Figure 2 shows a possible GOP of 9 frames (*pictures*) followed by the I frame of the subsequent GOP. Pictures consist of collections of macroblocks called *slices*. The "headers" shown at each level contain parameters relevant to each bitstream type, but no actual image data.

2.2 Experimental Model

For experimental purposes, we focus on a simplified machine model, limiting the microprocessor to a single level of caching, and assume that MPEG-2 decoding is the only active process. Our results come from trace-driven cache simulations of video stream decoding. The clips themselves are progressive MPEG-2 Main Level sequences, with a resolution of 704 × 480 pixels at 30 frames/second. The decoder is a version of mpeg2decode from the MPEG Software Simulation Group [1], running on a SPARC processor. We have modified the program to instrument it for trace generation with QPT2 (the Quick Profiler and Tracer) [4], a component of the Wisconsin Architectural Research Toolkit (WARTS). Our cache simulator, Pacino, is designed to support continuous media applications.

To limit the memory storage and CPU cycle requirements of simulation, we restrict each run to one GOP, the largest logical bitstream unit below a sequence. Simulations run as if at steady state – i.e. preceded and followed by a large number of other GOP's. This eliminates the side effects of program startup and termination.

Decoded video is sent to the frame buffer in 4:2:0 YCrCb form, the native representation of MPEG, rather than 4:4:4 RGB. YCrCb is a color representation with one luminance (grayscale) and two chrominance components per pixel, rather than one each of red, green, and blue. In 4:2:0 YCrCb, the chrominance components are subsampled in both dimensions. We assume that upsampling, color space conversion, and any dithering are done on the fly by the video display subsystem. These features are becoming common in workstations and budget PC video accelerators alike. According to prior work on MPEG-1 decoders, this processing can account for almost 25% of total execution time if performed in software [6]. In addition, the 4:2:0 representation of an image is only half the size of the 4:4:4 version, reducing the minimum sustained system bus load to 31.5 Mbytes/s.

3 Data Types and Definitions

The data set of an MPEG-2 decoder – the information that must be available to the program in the course of its execution – consists of several different types which serve distinct functions. As a result, they are quite heterogeneous with respect to the types and patterns of access utilized by the decoder, as well as the amount of memory space required. The following classes account for the global and static local values in user space.

- *Input.* The compressed MPEG-2 data stream, with data read in series from a fixed-size buffer and refreshed by system calls.
- *Output.* Uncompressed picture data in YCrCb format, stored in a video window image buffer. This data type is *write only* – written but never read by the CPU. System calls transfer each completed picture into the frame buffer.
- *Tabular.* Static, *read only* information used in the MPEG decoding process, such as various lookup tables.
- *Reference.* The current frame and the previously decoded frames used to reconstruct it (up to two for a B frame), in YCrCb form.
- *Block.* The DCT coefficient and pixel values for a single macroblock.
- *State.* Values incidental to the settings and operation of the decoder, yet not part of the image data *per se*.

This partitioning of data types may not be explicit within the decoder source, but represent an abstraction at a higher level of the data needed for decoder operation. Most of these data types are both read and written. The exceptions are the tabular data, which are *read only*, and the the output data which is *write only*. Within a memory hierarchy, reading and writing are not symmetric operations. Read misses and write misses in caches have very different latencies. The fact that some MPEG decoder data types are only read or written can be exploited for performance optimization.

Data Type	*Access Type*	*Size*	*Fraction of References*
Input	read/write	2 KB	2.7%
Output	write only	500 KB	3.9%
Tabular	read only	5 KB	5.5%
Reference	read/write	1500 KB	23.7%
Block	read/write	1.5 KB	31.4%
State	read/write	0.5 KB	25.6%

Table 1: Summary of decoder data types, sizes, access types, and proportion of memory references accounted for

3NSDOCID: <XP_____10251768A__I_>

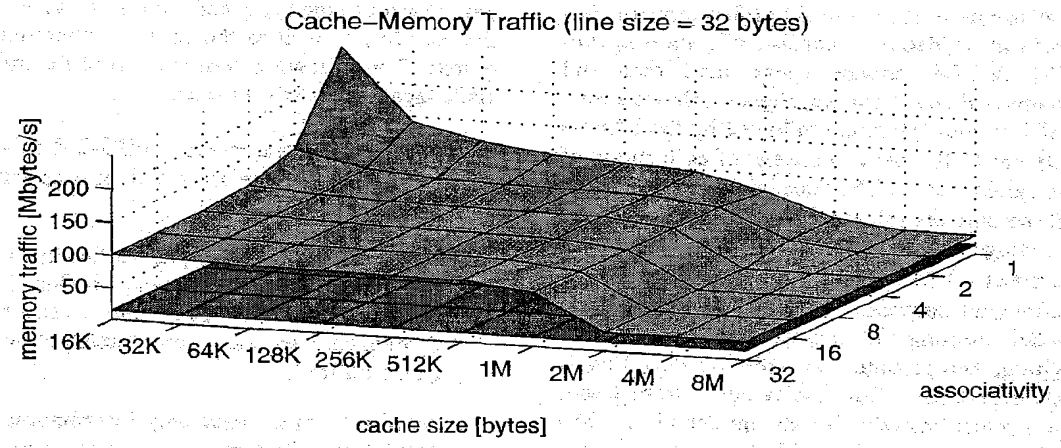


Figure 4: Typical cache-memory traffic for MPEG-2 Main Level decoding, over minimum possible traffic level

The essential properties of the different data types are summarized in Table 1. Note how in terms of size, the reference and output types completely dominate the others. With respect to caching, this means that their presence will tend to repeatedly expel the other types from the cache, except for very large data caches. It doesn't matter how many times the other values are re-used. Capacity limitations alone will insure that reference and output data, when updated, will throw the other data types out.

Curiously, ranking the data types in terms of the number of references rather than by size gives a very different picture. Block and state data account for by far the most memory requests. Reference data is next, but output data is in next to last place, accounting for 8 times fewer memory requests than block data. Note that the percentages don't add up to 100%; approximately 7.2% of data references are due to temporary variables and library functions.

4 Cache Simulation Results

Trace-driven cache simulations clarify how MPEG-2 decoder data requests translate into what traffic is seen by the memory. First of all, there is a distinct (albeit limited) benefit to caching video decoder data, even using naive, generic schemes. It has occasionally been suggested that caches are critically inefficient for video data. Accordingly, some special-purpose systems dispense with data caches altogether in favor of large register files [3]. Yet we have found that despite the large set sizes and the magnitude of data consumed and discarded, there is sufficient re-use of values for caching to significantly reduce the required memory bandwidth.

For example, decoding one Group Of Pictures from a typical MPEG-2 stream generates 2.1×10^8 memory requests. A 16 Kbyte, direct-mapped cache generates 3.9×10^7 words of memory traffic. If we estimate conservatively that each word of cache data represents a memory request, then even a relatively small, simple cache keeps almost 85% of traffic off of the memory bus. In fact, caches typically load several words at a time, while many CPU memory requests are for data types smaller than a word. Therefore, data caches can make far more efficient use of available memory bandwidth than registers alone. Even very clever register management would have trouble competing with anything but the smallest caches. The remainder of this section examines in more detail how the basic parameters of a simple cache affect memory traffic as well as misses.

4.1 Cache Size

The size of a cache is its most significant design parameter, certainly from a cost standpoint. Because cache size usually increases or decreases by factors of two, the decision of how large a cache to implement in a system is pivotal. For Main Level MPEG-2 decoding, the cache-memory traffic as a function of cache size and associativity shows little variation between video sequences. Figure 4 shows a representative plot, assuming a standard line size of 32 bytes. This is superimposed on a surface showing the minimum possible cache-memory traffic for the sequence, consisting of the encoded and decoded video streams combined.

The most prominent feature of the traffic function is a large plateau, at a level of 6.3 times the minimum value. Except for direct-mapped caches, which produce a peak of 16 times the minimum, cache-memory traffic changes little with increasing cache size for most of the range. Traffic starts to roll off at 1 Mbyte, and plummets at 2 Mbytes. This reflects the 2 Mbyte size of the decoder data set. Larger caches show negligible improvement, with the ad-

BNSDOCID: <XP_____10251768A__I_>

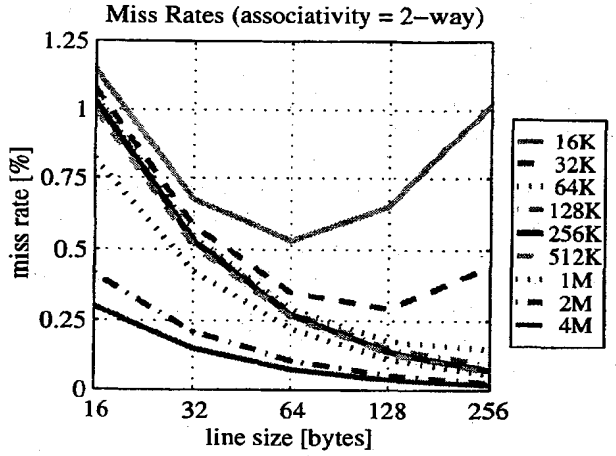


Figure 5: Typical miss rates for MPEG-2 Main Level decoding over different cache sizes

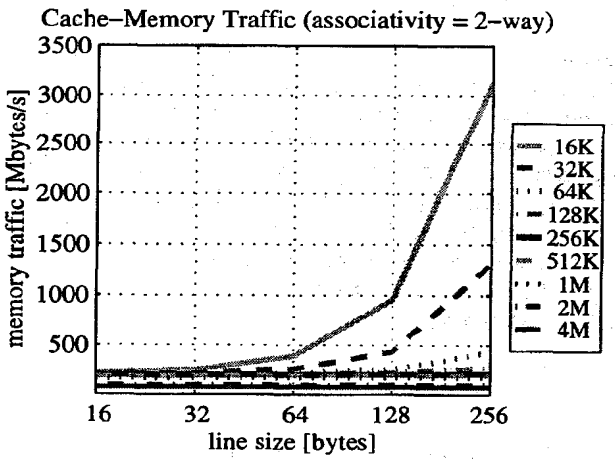


Figure 6: Typical cache-memory traffic for MPEG-2 Main Level decoding over different cache sizes

ditional space providing no extra benefit. However, the measured minimum is still almost 2 times higher than the minimum possible value.

These data imply that a 32 Kbyte cache is just as good (or bad) for MPEG-2 Main Level decoding as a 64 Kbyte or 128 Kbyte one, for example. Improving on cache-memory traffic requires a 1 Mbyte cache or higher, but a 2 Mbytes or larger cache only brings traffic down to double the absolute minimum, leaving plenty of room for improvement.

4.2 Associativity

Increasing set associativity is a popular method for getting higher performance out of smaller caches. The memory traffic effects of varying set associativity are also visible in Figure 4. Going from a direct-mapped cache to a 2-way set-associative one can reduce memory traffic by as much as 50% for small caches. Increasing associativity to 4 can squeeze out almost another 10% improvement over the direct-mapped case. Set sizes of greater than 4, however, show minimal benefit across all cache sizes. Since increasingly higher levels of associativity add considerably to cache cost, complexity and access time, such enhancements are not justified.

While memory traffic is the primary focus, we would prefer to not adversely affect miss rates. As a function of cache size and set associativity, miss rates show the same behavioral pattern as memory traffic. However, unlike for cache-memory traffic, the performance is more than satisfactory, with an average value below 0.5%.

4.3 Line Size

Line size, another fundamental parameter, is less costly to experiment with than cache size. Subblock placement can help decouple the size of cache lines and that of the memory bus. Unfortunately, there is contention between miss rate and memory traffic minimization. Low miss rates call for larger lines than the typical 32 bytes, as illustrated in Figure 5. Larger lines tend to provide superior spatial locality, but require more data to be read and possibly written back on a miss. For this reason, minimal memory traffic occurs with the smaller lines. This relationship is readily apparent in Figure 6, where for the smallest caches, the largest line sizes lead to cache-memory traffic almost 200 times the absolute minimum.

Balancing the demands of miss rates and memory traffic requires further investigation. For the time being, maintaining the ubiquitous 32 byte line size seems sensible. The interests of memory traffic reduction argue against a switch to larger values.

5 Architectural Solutions

It is clear that conventional cache techniques provide limited relief from excess cache-memory traffic. The simplest way to improve memory traffic and miss rates is to have a very large cache – but a large cache is not always feasible, and simulations show that improving MPEG-2 decoding performance requires very big caches. In any case, one would prefer to extract better performance from smaller, lower-cost resources. Hardware prefetching has been suggested as a remedy [9], yet primarily as a means to reduce miss rates and without much consideration of cache-memory traffic, which tends to increase with prefetching.

We advocate looking more closely at the internal dynamics of the decoder itself. Exploiting knowledge of individual data types, their sizes, access types, and access

3NSDOCID: <XP_____10251768A__I_>

patterns can suggest architectural solutions to cache inefficiency. In particular, there are great potential benefits in treating different data types distinctly. Write-only values, like video output data, are clearly a waste of cache space. While block and state data account for most of the memory requests their relatively small size means that larger data types, like reference data, systematically and relentlessly eject them from the cache. Preventing blatant cache pollution and the predation of small data types with high re-use can yield considerable cache-memory traffic savings. Therefore, we are investigating the following traffic-reduction techniques.

- *Selective Caching.* Exclude specific data from the cache.
- *Line Locking.* Reserve selected cache lines for particular data objects; lines are untouched by cache replacement while locked.
- *Locality Hints.* Flag data with poor temporal locality to guide cache replacement policies.
- *Scratch Memory.* Implement a small portion of addressable system memory on the processor – not as cache – for storage of small, frequently-used data.

As an example, consider selective caching, its rationale and implementation. Data objects with no possibility of re-use, like video output data, steal cache space from other data. Bypassing the cache entirely in favor of direct storage to main memory promises more efficient use. The UltraSPARC has block load and store instructions, which move groups of data directly between registers and main memory. The PowerPC architecture provides for marking areas of memory (i.e. pages) as non-cacheable. Once marked, data is transparently and automatically transferred between registers and memory buffers; the block load/store approach requires more explicit programming effort.

Preliminary results from simulation are promising. Excluding video output data alone from the cache reduces cache-memory traffic by up to 50%. Across the configurations considered, not caching video output yields a 25% reduction on average. Improvements in cache-memory traffic are global, yet the shape of the curve is not substantially modified. The plateau out to 512 Kbytes persists, as does the swift drop down at 2 Mbytes. Excluding output data is even more helpful for miss rates, which drop by a maximum of 85% from earlier levels, and 60% on average.

6 CONCLUSIONS

This paper has explained the memory traffic and data cache behavior of MPEG-2 decoding on a general-purpose computer. We have shown that even simple caches play a key role in reducing the required memory bandwidth. However, manipulating conventional cache parameters like cache size, associativity, and line size has a limited potential to reduce cache-memory traffic. Increasing cache size is ineffectual until the 1 Mbyte threshold, and even extremely large caches still generate double the absolute minimum required traffic. Enhanced set associativity becomes increasingly cost-ineffective above 4-way implementations, and deviating from a 32-byte line size leads to either increased traffic or miss rates. However, when driven by an understanding of decoder behavior, even relatively simple architectural enhancements, like selective caching, can dramatically improve cache efficiency. We anticipate that further refinement of this knowledge will lead to original and cost-effective ways of improving the decoding performance of general purpose microprocessors, and accelerate the deployment of digital video platforms and applications.

References

[1] Stefan Eckart et al. ISO/IEC MPEG-2 software video codec. In *Proc. Digital Video Compression: Algorithms and Technologies 1995*, pages 100–109. SPIE, Jan. 1995.

[2] International Organization for Standardization. *ISO/IEC JTC1/SC29/WG11/602 13818-2 Committee Draft (MPEG-2)*, Nov. 1993.

[3] Paul Kalapathy. Hardware-software interactions on MPACT. *IEEE Micro*, 17(2):20–26, Mar./Apr. 1997.

[4] James R. Larus. Efficient program tracing. *IEEE Computer*, 26(5):52–61, May 1993.

[5] Ruby B. Lee. Subworld parallelism with MAX-2. *IEEE Micro*, 16(4):51–59, Aug. 1996.

[6] Ketan Patel et al. Performance of a software MPEG video decoder. In *Proc. 1st ACM Intl. Conf. on Multimedia*, pages 75–82. ACM, Aug. 1993.

[7] Alex Peleg et al. Intel MMX for multimedia PCs. *Comm. of the ACM*, 40(1):25–38, Jan. 1997.

[8] Marc Tremblay et al. VIS speeds new media processing. *IEEE Micro*, 16(4):10–20, Aug. 1996.

[9] Daniel F. Zucker et al. A comparison of hardware prefetching techniques for multimedia. Technical Report CSL-TR-95-683, Stanford University Depts. of Electrical Engineering and Computer Science, Stanford, CA, Dec. 1995.

BNSDOCID: <XP_____10251768A__I_>

Europäisches Patentamt
European Patent Office
Office européen des brevets

European Patent Office
80298 MUNICH
GERMANY
Tel. +49 (0)89 2399 - 0
Fax +49 (0)89 2399 - 4465

TAMPEREEN PATENTTITOIMISTO OY
01. 02. 2010

For any questions about this communication:
Tel.:+31 (0)70 340 45 00

Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy
Hermiankatu 1 B
33720 Tampere
FINLANDE

Date: 02.02.10

Reference	Application No./Patent No.
TP104815/EPG	06101014.6 - 2223 / 1670260
Applicant/Proprietor: Nokia Corporation	

Communication

The extended European search report is enclosed.

The extended European search report includes, pursuant to Rule 62 EPC, the European search report (R. 61 EPC) or the partial European search report/ declaration of no search (R. 63 EPC) and the European search opinion.

Copies of documents cited in the European search report are attached.

☑ 0 additional set(s) of copies of such documents is (are) enclosed as well.

The following have been approved:

☑ Abstract ☑ Title

☐ The Abstract was modified and the definitive text is attached to this communication.

The following figure(s) will be published together with the abstract: 1

Refund of the search fee

If applicable under Article 9 Rules relating to fees, a separate communication from the Receiving Section on the refund of the search fee will be sent later.

Europäisches Patentamt • European Patent Office • Office européen des brevets

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 06 10 1014

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	ELLOUMI O ET AL: "Issues in multimedia scaling: the MPEG video streams case" EMERGING TECHNOLOGIES AND APPLICATIONS IN COMMUNICATIONS, 1996. PROCEE DINGS., FIRST ANNUAL CONFERENCE ON PORTLAND, OR, USA 7-10 MAY 1996, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 7 May 1996 (1996-05-07), pages 46-49, XP010164870 ISBN: 978-0-8186-7585-0 * paragraph [0002] *	1-17	INV. H04N7/36
X	WENGER S ED - WONG P W ET AL: "Temporal scalability using P-pictures for low-latency applications" MULTIMEDIA SIGNAL PROCESSING, 1998 IEEE SECOND WORKSHOP ON REDONDO BEACH, CA, USA 7-9 DEC. 1998, PISCATAWAY, NJ, USA,IEEE, US, 7 December 1998 (1998-12-07), pages 559-564, XP010318327 ISBN: 978-0-7803-4919-3 * the whole document *	1-17	
P,X	HANNUKSELA: "Enhanced Concept of GOP" JOINT VIDEO TEAM (JVT) OF ISO/IEC MPEG & ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 AND ITU-T SG16 Q6), XX, XX, no. JVT-B042r1, 1 February 2002 (2002-02-01), XP030005042 * the whole document *	1-17	TECHNICAL FIELDS SEARCHED (IPC) H04N
A	WO 01/47283 A (GEN INSTRUMENT CORP [US]) 28 June 2001 (2001-06-28) * abstract *	1-17	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	22 January 2010	Wahrenberg, Annika

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 06 10 1014

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

22-01-2010

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 0147283 A	28-06-2001	AU 4338800 A	03-07-2001
		CA 2395605 A1	28-06-2001
		CN 1435055 A	06-08-2003
		DE 60009159 D1	22-04-2004
		DE 60009159 T2	25-11-2004
		EP 1249131 A1	16-10-2002
		TW 465243 B	21-11-2001

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

The examination is being carried out on the **following application documents**:

Description, Pages
1-33 as originally filed
Claims, Numbers
1-17 as originally filed
Drawings, Sheets
1/4-4/4 as originally filed

1. Reference is made to the following document; the numbering will be adhered to in the rest of the procedure:

   D1: ELLOUMI O ET AL: "Issues in multimedia scaling: the MPEG video streams case" EMERGING TECHNOLOGIES AND APPLICATIONS IN COMMUNICATIONS, 1996. PROCEE DINGS., FIRST ANNUAL CONFERENCE ON PORTLAND, OR, USA 7-10 MAY 1996, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 7 May 1996 (1996-05-07), pages 46-49, XP010164870 ISBN: 978-0-8186-7585-0

2. The present application is a divisional application of EP03700321 (from here on called the parent). In accordance with Article 76(1) EPC, the subject matter of a divisional application may not extend beyond the content of the parent application as filed. The present application adds subject-matter which goes beyond the parent application as filed, the reasons being as follows:

   The claims and the description of the parent application are directed to scalable coding of a video sequence, the video sequence being formed of a number of sub-sequences. Furthermore, the following feature is described as essential in the parent application: inserting information into the bitstream describing the interdependencies of the sub-sequences, i.e. information pertaining to which sub-sequence each subsequence has been predicted from (see. para. [0017] and [0033]).

   Present claim 1 relates to a method of encoding a video sequence, including encoding an indication of at least one image frame which is the first image frame, in decoding order, of an independent sequence. There is no mention of scalable coding or information describing the dependencies among different sub-

sequences. The same holds for claims 5, 9, 10, 13, 14 and 16.

The examiner finds no basis for this generalisation in the parent application as filed, whether in the claims or in the description. Features which were described as essential in the parent are not present in the divisional. Thus, subject-matter has been added with respect to the parent, contrary to Article 76(1) EPC.

3. In view of the above objections it is not at present practicable to carry out a full examination of the subject-matter of the claims. The applicant is therefore requested to file amended claims on which the further prosecution of the application is to be based. The applicant should indicate the passages of the parent application as filed on which the new claims are based.

4. It is also worth mentioning that claim 1 is not new (Article 52(1) and 54(1) and (2) EPC) with respect to an MPEG coded stream. An MPEG stream consists of a number of independent sequences called GOPs which start with an intra-frame (I) followed by a number of predicted frames (P) (see D1, para. 2). In the bitstream, there is an indication of the frame type (I or P) which, in the case of an I frame, identifies the start of an independent sequence. The same reasoning applies for claims 5, 9, 10, 13, 14 and 16, which are therefore not new either (Article 52(1) and 54(1) and (2) EPC).

***

Europäisches Patentamt
European Patent Office
Office européen des brevets

European Patent Office
80298 MUNICH
GERMANY
Tel. +49 (0)89 2399 - 0
Fax +49 (0)89 2399 - 4465

TAMPEREEN PATENTTITOIMISTO OY
01. 02. 2010

Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy
Hermiankatu 1 B
33720 Tampere
FINLANDE

For any questions about this communication:
Tel.:+31 (0)70 340 45 00

Date: 02.02.10

Reference	Application No./Patent No.
TP104814/EPG	06101013.8 - 2223 / 1670259

Applicant/Proprietor: Nokia Corporation

Communication

The extended European search report is enclosed.

The extended European search report includes, pursuant to Rule 62 EPC, the European search report (R. 61 EPC) or the partial European search report/ declaration of no search (R. 63 EPC) and the European search opinion.

Copies of documents cited in the European search report are attached.

☑ 0 additional set(s) of copies of such documents is (are) enclosed as well.

The following have been approved:

☑ Abstract ☑ Title

☐ The Abstract was modified and the definitive text is attached to this communication.

The following figure(s) will be published together with the abstract: 1

Refund of the search fee

If applicable under Article 9 Rules relating to fees, a separate communication from the Receiving Section on the refund of the search fee will be sent later.

Europäisches Patentamt • European Patent Office • Office européen des brevets

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 06 10 1013

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	WENGER S ED - WONG P W ET AL: "Temporal scalability using P-pictures for low-latency applications" MULTIMEDIA SIGNAL PROCESSING, 1998 IEEE SECOND WORKSHOP ON REDONDO BEACH, CA, USA 7-9 DEC. 1998, PISCATAWAY, NJ, USA,IEEE, US, 7 December 1998 (1998-12-07), pages 559-564, XP010318327 ISBN: 978-0-7803-4919-3 * the whole document *	1-31	INV. H04N7/36
A	SODERQUIST P ET AL: "Memory traffic and data cache behavior of an MPEG-2 software decoder" COMPUTER DESIGN: VLSI IN COMPUTERS AND PROCESSORS, 1997. ICCD '97. PRO CEEDINGS., 1997 IEEE INTERNATIONAL CONFERENCE ON AUSTIN, TX, USA 12-15 OCT. 1997, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 12 October 1997 (1997-10-12), pages 417-422, XP010251768 ISBN: 978-0-8186-8206-3 * the whole document *	1-31	TECHNICAL FIELDS SEARCHED (IPC) H04N
P,X	HANNUKSELA: "Enhanced Concept of GOP" JOINT VIDEO TEAM (JVT) OF ISO/IEC MPEG & ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 AND ITU-T SG16 Q6), XX, XX, no. JVT-B042r1, 1 February 2002 (2002-02-01), XP030005042 * the whole document *	1-31	
A	EP 0 898 427 A (MITSUBISHI ELECTRIC CORP [JP]) 24 February 1999 (1999-02-24) * abstract *	1-31	
A	WO 01/47283 A (GEN INSTRUMENT CORP [US]) 28 June 2001 (2001-06-28) * abstract *	1-31	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	22 January 2010	Wahrenberg, Annika

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 06 10 1013

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

22-01-2010

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0898427 A	24-02-1999	CN 1217852 A	26-05-1999
		CN 1510928 A	07-07-2004
		CN 101005624 A	25-07-2007
		CN 1992900 A	04-07-2007
		CN 101026762 A	29-08-2007
		WO 9836576 A1	20-08-1998
		JP 3748895 B2	22-02-2006
		JP 2009071877 A	02-04-2009
		JP 2009071878 A	02-04-2009
		JP 2009071879 A	02-04-2009
		US 2002114392 A1	22-08-2002
WO 0147283 A	28-06-2001	AU 4338800 A	03-07-2001
		CA 2395605 A1	28-06-2001
		CN 1435055 A	06-08-2003
		DE 60009159 D1	22-04-2004
		DE 60009159 T2	25-11-2004
		EP 1249131 A1	16-10-2002
		TW 465243 B	21-11-2001

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

The examination is being carried out on the **following application documents**:

Description, Pages
1-33 as originally filed
Claims, Numbers
1-31 as originally filed
Drawings, Sheets
1/4-4/4 as originally filed

1. Reference is made to the following document; the numbering will be adhered to in the rest of the procedure:

   D1: SODERQUIST P ET AL: "Memory traffic and data cache behavior of an MPEG-2 software decoder" COMPUTER DESIGN: VLSI IN COMPUTERS AND PROCESSORS, 1997. ICCD '97. PRO CEEDINGS., 1997 IEEE INTERNATIONAL CONFERENCE ON AUSTIN, TX, USA 12-15 OCT. 1997, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 12 October 1997 (1997-10-12), pages 417-422, XP010251768 ISBN: 978-0-8186-8206-3

2. The present application has six independent claims. Under Rule 43(2) EPC, an application may not contain more than one independent claim in a particular category unless the subject-matter claimed falls within one or more of the exceptional situations set out in sub-paragraph (a), (b) or (c).

   None of the exceptions apply in the present case. Although separate claims to an encoder and the corresponding decoder is allowed (Rule 43(2)(a) EPC), encoder claim 26 does not mirror the features of decoder claim 10. The same holds for device claim 28 and signal claim 30. The scope the independent claims varies slightly but there is a significant degree of overlap, which makes the claims as a whole unclear (Article 84 EPC). It is not even apparent that the claims are unified by a single general inventive concept.

3. The present application is a divisional application of EP03700321 (from here on called the parent). In accordance with Article 76(1) EPC, the subject matter of a divisional application may not extend beyond the content of the parent application as filed. The present application adds subject-matter which goes beyond the parent application as filed, the reasons being as follows:

The claims and the description of the parent application are directed to scalable coding of a video sequence, the video sequence being formed of a number of sub-sequences. Furthermore, the following feature is described as essential in the parent application: inserting information into the bitstream describing the interdependencies of the sub-sequences, i.e. information pertaining to which sub-sequence each subsequence has been predicted from (see. para. [0017] and [0033]).

Present claim 1 relates to the decoding of a video sequence where there is a discontinuity of numbering of the frames. There is no mention of sub-sequences and their relationship relative each other. Claim 1 is not even limited to predictive coding, let a lone scalable coding.

In Claims 26 and 28, the discontinuity is defined as frames being excluded from the video sequence. There is no mention of dependent sub-sequences or scalable coding.

The examiner finds no basis for this generalisation in the parent application as filed, whether in the claims or in the description. Features which were described as essential in the parent are not present in the divisional. Thus, subject-matter has been added with respect to the parent, contrary to Article 76(1) EPC.

4. In view of the above objections it is not at present practicable to carry out a full examination of the subject-matter of the claims. The applicant is therefore requested to file amended claims on which the further prosecution of the application is to be based. The applicant should indicate the passages of the parent application as filed on which the new claims are based.

5. It is also worth mentioning that claim 1 is not new (Article 54(1) and (2) EPC) with respect to standard MPEG decoding. The discontinuity in claim 1 covers B-frame decoding where the frames are processed in a non-temporal order (see e.g. D1, fig. 2 and the associated text). The requirements of Article 52(1) EPC are, therefore, not met.

***

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 898 427 A1

(12) EUROPEAN PATENT APPLICATION

published in accordance with Art. 158(3) EPC

(43) Date of publication:
24.02.1999 Bulletin 1999/08

(21) Application number: **98900693.7**

(22) Date of filing: **22.01.1998**

(51) Int. Cl.[6]: **H04N 7/32**

(86) International application number:
PCT/JP98/00232

(87) International publication number:
WO 98/36576 (20.08.1998 Gazette 1998/33)

(84) Designated Contracting States:
DE FR

(30) Priority: **13.02.1997 JP 28862/97**

(71) Applicant:
MITSUBISHI DENKI KABUSHIKI KAISHA
Tokyo 100 (JP)

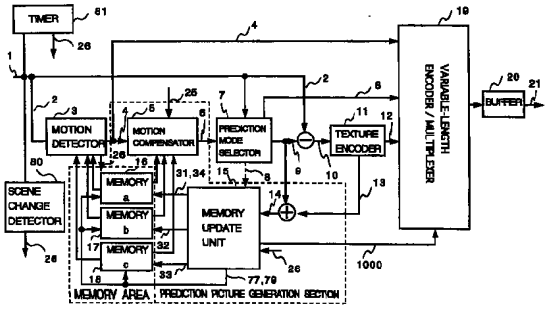
(72) Inventors:
- **SEKIGUCHI, Shunichi,**
  Mitsubishi Denki K.K.
  Chiyoda-ku Tokyo 100 (JP)
- **ASAI, Kohtaro,**
  Mitsubishi Denki K.K.
  Chiyoda-ku Tokyo 100 (JP)
- **MURAKAMI, Tokumichi,**
  Mitsubishi Denki K.K.
  Chiyoda-ku Tokyo 100 (JP)
- **NISHIKAWA, Hirofumi,**
  Mitsubishi Denki K.K.
  Chiyoda-ku Tokyo 100 (JP)
- **KURODA, Shinichi,**
  Mitsubishi Denki K.K.
  Chiyoda-ku Tokyo 100 (JP)
- **ISU, Yoshimi,**
  Mitsubishi Denki K.K.
  Chiyoda-ku Tokyo 100 (JP)
- **HASEGAWA, Yuri,**
  Mitsubishi Denki K.K.
  Chiyoda-ku Tokyo 100 kyo 100 (JP)

(74) Representative:
Pfenning, Meinig & Partner
Mozartstrasse 17
80336 München (DE)

(54) MOVING IMAGE ESTIMATING SYSTEM

(57) To achieve an encoding system including a highly efficient prediction performed in response to the content of a scene, a significance, and a motion characteristic of a moving picture and the like, memories a, b, c, motion compensator 5 responsive to an arbitrary transform parameter representing the motion of a prediction picture segment for generating a predicted picture by using arbitrary data stored in the memories a, b, c based upon the transform parameter, and memory update unit 15 for allowing the content of one or more of the memories to be updated at an arbitrary period of time, are provided.

Fig.1



EP 0 898 427 A1

Description

Technical Field

[0001] The present invention relates to the prediction of a moving picture implemented, for example, in

a moving picture encoder/decoder used in a portable/stationary video communication device and the like for visual communications in a video telephone system, a video conference system or the like,
a moving picture encoder/decoder used in a picture storage/recording apparatus such as a digital VTR and a video server, and
a moving picture encoding/decoding program implemented in the form of a single software or a firmware as a Digital Signal Processor (DSP).

Background Art

[0002] MPEG-4 (Moving Picture Experts Group Phase-4) Video Encoding/Decoding Verification Model (hereinafter referred to by the initials VM) whose standardization is in progress by ISO/IEC JTC1/SC29/WG11 may be introduced as a conventional type of predictive encoding/decoding in an encoding/decoding system of moving pictures. The VM continues to revise its contents according to the progress being made in standardization of MPEG-4. Here, Version 5.0 of the VM is designated to represent the VM and will be simply referred to as VM hereinafter.

[0003] The VM is a system for encoding/decoding each video object as one unit in view of a moving picture sequence being an aggregate of video objects changing their shapes time-/space-wise arbitrarily. Fig. 29 shows a VM video data structure. According to the VM, a time-based moving picture object is called a Video Object (VO), and picture data representing each time instance of the VO, as an encoding unit, is called a Video Object Plane (VOP). If the VO is layered in time/space, a special unit called a Video Object Layer (VOL) is provided between the VO and the VOP for representing a layered VO structure. Each VOP includes shape information and texture information to be separated. If the moving picture sequence includes a single VO, then the VOP is equated to a frame. There is no shape information included, in this case, and the texture information alone is then to be encoded/decoded.

[0004] The VOP includes alpha data representing the shape information and texture data representing the texture information, as illustrated in Fig. 30. Each data are defined as an aggregate of blocks (alphablocks/macroblocks), and each block in the aggregate is composed of 16x16 samples. Each alphablock sample is represented in eight bits. A macroblock includes accompanied chrominance signals being associated with 16x16 sample luminance signals. VOP data are obtained from a moving picture sequence externally processed outside of an encoder.

[0005] Fig. 31 is a diagram showing the configuration of a VOP encoder according to the VM encoding system. The diagram includes original VOP data P1 to be inputted, an alphablock P2 representing the shape information of the VOP, a switch P3a for passing the shape information, if there is any, of the inputted original VOP data, a shape encoder P4 for compressing and encoding the alphablock, compressed alphablock data P5, a locally decoded alphablock P6, texture data (a macroblock) P7, a motion detector P8, a motion parameter P9, a motion compensator P10, a predicted picture candidate P11, a prediction mode selector P12, a prediction mode P13, a predicted picture P14, a prediction error signal P15, a texture encoder P16, texture encoding information P17, a locally decoded prediction error signal P18, a locally decoded macroblock P19, a sprite memory update unit P20, a VOP memory P21, a sprite memory P22, a variable-length encoder/multiplexer P23, a buffer P24, and an encoded bitstream P25.

[0006] Fig. 32 shows a flowchart outlining an operation of the encoder.

[0007] Referring to the encoder of Fig. 31, the original VOP data P1 are decomposed into the alphablocks P2 and the macroblocks P7 (Steps PS2 and PS3). The alphablocks P2 and the macroblocks P7 are transferred to the shape encoder P4 and the motion detector P8, respectively. The shape encoder P4 is a processing block for data compression of the alphablock P2 (step PS4), the process of which is not discussed here further in detail because the compression method of shape information is not particularly relevant to the present invention.

[0008] The shape encoder P4 outputs the compressed alphablock data P5 which is transferred to the variable-length encoder/multiplexer P23, and the locally decoded alpha data P6 which is transferred sequentially to the motion detector P8, the motion compensator P10, the prediction mode selector P12, and the texture encoder P16.

[0009] The motion detector P8, upon reception of the macroblock P7, detects a local-motion vector on a macroblock basis using reference picture data stored in the VOP memory P21 and the locally decoded alphablock P6 (step PS5). Here, the motion vector is one example of a motion parameter. The VOP memory P21 stores the locally decoded picture of a previously encoded VOP. The content of the VOP memory P21 is sequentially updated with the locally decoded picture of a macroblock whenever the macroblock is encoded. In addition, the motion detector P8 detects a global warping parameter, upon reception of the full texture data of the original VOP, by using reference picture data stored in the sprite memory P22 and locally decoded alpha data. The sprite memory P22 will be discussed later in detail.

[0010] The motion compensator P10 generates the predicted picture candidate P11 by using the motion

parameter P9, which is detected in the motion detector P8, and the locally decoded alphablock P6 (step PS6). Then, the prediction mode selector P12 determines the final of the predicted picture P14 and corresponding prediction mode P13 of the macroblock by using a prediction error signal power and an original signal power (step PS7). In addition, the prediction mode selector P12 judges the coding type of the data either intra-frame coding or inter-frame coding.

[0011] The texture encoder P16 processes the prediction error signal P15 or the original macroblock through Discrete Cosine Transformation (DCT) and quantization to obtain a quantized DCT coefficient based upon the prediction mode P13. An obtained quantized DCT coefficient is transferred, directly or after prediction, to the variable-length encoder/multiplexer P23 to be encoded (steps PS8 and PS9). The variable-length encoder/multiplexer P23 converts the received data into a bitstream and multiplexes the data based upon predetermined syntaxes and variable-length codes (step PS10). The quantized DCT coefficient is subject to dequantization and inverse DCT to obtain the locally decoded prediction error signal P18, which is added to the predicted picture P14, and the locally decoded macroblock P19 (step PS11) is obtained. The locally decoded macroblock P19 is written into the VOP memory P21 and the sprite memory P22 to be used for a later VOP prediction (step PS12).

[0012] Dominant portions of prediction including a prediction method, a motion compensation, and the update control of the sprite memory P22 and the VOP memory P21 will be discussed below in detail.

(1) Prediction method in the VM

[0013] Normally, four different types of VOP encoding shown in Fig. 33 are processed in the VM. Each encoding type is associated with a prediction type or method marked by a circle on a macroblock basis. With an I-VOP, intra-frame coding is used singly involving no prediction. With a P-VOP, past VOP data can be used for prediction. With a B-VOP, both past and future VOP data can be used for prediction.

[0014] All the aforementioned prediction types are motion vector based. On the other hand, with a Sprite-VOP, a sprite memory can be used for prediction. The sprite is a picture space generated through a step-by-step mixing process of VOPs based upon a warping parameter set

$$\vec{a}=(a, b, c, d, e, f, g, h)$$

detected on a VOP basis (The mark $\rightarrow$ denotes a vector hereinafter). The warping parameter set is determined by the following parametric equations.

$$x'=(ax+by+c)/(gx+hy+1)$$

$$y'=(dx+ey+f)/(gx+hy+1)$$

The sprite is stored in the sprite memory P22.

[0015] Referring to the parametric equations, (x, y) represents the pixel position of an original VOP in a two-dimensional coordinate system. (x', y') represents a pixel position in the sprite memory corresponding to (x, y,) based upon a warping parameter. With the Sprite-VOP, the warping parameter set can be used uniformly with each macroblock to determine (x', y') in the sprite memory for prediction to generate a predicted picture. In a strict sense, the sprite includes "Dynamic Sprite" used for prediction and "Statistic Sprite" used for prediction as well as for another purpose of an approximate representation of VOP at a decoding station. In Figs. 34 through 37 below, "sprite" stands for Dynamic Sprite.

[0016] The motion detector P8 detects the motion vector and the warping parameter to be used for the aforementioned prediction types. The motion vectors and the warping parameters are generically called the motion parameter P9 hereinafter.

(2) Motion compensation

[0017] Fig. 34 is a diagram showing the configuration of the motion compensator P10 in detail. In the figure, a warping parameter P26, a motion vector P27, a global-motion compensator P28, a local-motion compensator P29, a warping-parameter based predicted picture candidate P30, and a motion-vector based predicted picture candidate P31 are shown. The warping-parameter and motion-vector based predicted picture candidates 30, 31 are generically called the predicted picture candidates P11 hereinafter.

[0018] Fig. 35 shows a flowchart outlining the operation of the motion compensator P10 including steps PS14 through PS21.

[0019] The motion compensator P10 generates the predicted picture candidate P11 using the warping parameter P26 of a full VOP detected on a macroblock P7 basis in the motion detector P8 or a macroblock based motion vector P27. The global-motion compensator P28 performs a motion compensation using the warping parameter P26, and the local-motion compensator P29 performs a motion compensation using the motion vector P27.

[0020] With the I-VOP, the motion compensator P10 does not operate. (The operating step proceeds to step PS21 from step PS14.) With a VOP other than the I-VOP, the local-motion compensator P29 reads out a predicted picture candidate PR1 from the locally decoded picture of a past VOP stored in the VOP memory P21 by using the motion vector P27 (step PS15). With the P-VOP, the predicted picture candidate PR1 is only available to be used.

[0021] When the B-VOP is identified in step PS16, the local-motion compensator P29 further reads out a predicted picture candidate PR2 from the locally decoded

picture of a future VOP stored in the VOP memory P21 by using the motion vector P27 (step PS17). In addition, an arithmetic mean of the predicted picture candidates PR1, PR2 obtained from the past and future VOP locally decoded pictures to obtain a predicted picture candidate PR3 (step PS18).

[0022] A predicted picture candidate PR4 is generated also through Direct Prediction (step PS19). (Direct Prediction is based upon a prediction method corresponding to B-Frame in an encoding method H.263, Recommendation ITU-T. A vector for B-Frame is produced based upon a group of P-VOP vectors, which is not discussed further here in detail.) In Fig. 34, the motion-vector based predicted picture candidates P31 is a generic term for all or part of the predicted picture candidates PR1 through PR4.

[0023] If a VOP is of neither I-VOP nor B-VOP, then the VOP is of Sprite-VOP. With the Sprite-VOP, the predicted picture candidate PR1 is read out from the VOP memory based upon the motion vector. In addition, the global-motion compensator P28 reads out the predicted picture candidate P30 from the sprite memory P22 based upon the warping parameter P26 in step PS20.

[0024] The global-motion compensator P28 calculates the address of a predicted picture candidate in the sprite memory P22 based upon the warping parameter P26, and reads out the predicted picture candidate P30 from the sprite memory P22 to be outputted based upon a resultant address. The local-motion compensator P29 calculates the address of a predicted picture candidate in the VOP memory P21 based upon the motion vector P27 and reads out the predicted picture candidate P31 to be outputted based upon a resultant address.

[0025] These predicted picture candidates P11 are evaluated along with an intra-frame coding signal of the texture data P7 in the prediction mode selector P12, which selects a predicted picture candidate having the least power of a prediction error signal along with a prediction mode.

(3) Updating of memories

[0026] The memory update unit P20 controls the VOP memory P21 and sprite memory P22 to be updated (step PS12). The contents of these memories are updated regardless of the prediction mode P13 selected on a macroblock basis.

[0027] Fig. 36 is a diagram showing the configuration of the memory update unit P20. Fig. 37 shows a flowchart including steps PS22 through PS28 illustrating the operation of the memory update unit P20.

[0028] In Fig. 36, an externally supplied VOP encoding type P32, an externally supplied sprite prediction identification flag P33 for indicating the use of the sprite memory for prediction, an externally supplied blend factor P34 used for prediction with the sprite memory, switches P35, P36, a sprite blender P37, a sprite transformer P38, a VOP memory update signal P39, and a sprite update signal P40 are shown.

[0029] Firstly, the use of the sprite with the current VO or VOL is examined if being designated by the sprite prediction identification flag P33 (step PS22). With no use of the sprite designated, the data are examined if being the B-VOP (step PS27). With the B-VOP, then no updating is performed with the VOP memory P21. With either the I-VOP or the P-VOP, then the VOP memory P21 is written over with the locally decoded macroblock P19 on a macroblock basis (step PS28).

[0030] With the use of the sprite designated in step PS22, then the VOP memory P21 is updated in the same manner as above (steps PS23, PS24), and in addition, the sprite memory PS22 is updated through the following procedure.

a) Sprite warping (step PS25)

[0031] In the sprite transformer P38, an area

$$M(\vec{R},t-1)$$

in the sprite memory P22 ($M(\vec{R},t-1)$ is an area having the same size as that of a VOP having the origin of the coordinates at a position in the sprite memory P22 with the VOP at a time t) is subject to warping (transformation) based upon a warping parameter

$$\vec{\alpha}=(a,b,c,d,e,f,g,h).$$

b) Sprite blending (step PS26)

[0032] By using a resultant warped picture from a) above, a new sprite memory area is calculated in the sprite blender P37 according to the following expression,

$$M(\vec{R},t)=(1-\alpha)\cdot W_b[M(\vec{R},t-1),\vec{\alpha}]+\alpha\cdot VO(\vec{r},t),$$

where α is the blend factor P34, $W_b[M,\vec{\alpha}]$ is the resultant warped picture, and $VO(\vec{r},t)$ is a pixel value of a locally decoded VOP with a location $\vec{r}$ and a time t.

[0033] With a non-VOP area in a locally decoded macroblock, it is assumed that

$$VO(\vec{r},t)=0.$$

As the blend factor α is assigned on a VOP basis, a locally decoded VOP is collectively blended into the sprite memory P22 based upon a weight α, regardless of the contents of a VOP area.

[0034] According to the aforementioned prediction system in the conventional encoding system, the video object is predicted by using the memory designed to be used for detecting the motion vector alone and the memory designed to be used for detecting the warping parameter alone, both of which are structurally allowed the maximum use of a single screen alone each. Thus,

the limited use of reference pictures is only available for prediction, thereby hindering a sufficient improvement in prediction efficiency.

[0035] Further, in such a system where two or more video objects are encoded concurrently, these memories only include a reference picture representing the past record of a video object to be predicted alone, which limits the variation of a reference picture and precludes the utilization of a correlation among video objects for prediction.

[0036] Further, the memories are updated regardless of such items as the internal structure, a characteristic, and the past record of the video object. This results in the insufficient storage of information lacking significant data for predicting a video object, thereby posing a problem of failing to enhance prediction efficiency.

[0037] The present invention is directed to solving the aforementioned problems. An objective of this invention is to provide the prediction system for encoding/decoding of picture data where two or more memories are provided to store the past record of the moving picture sequence effectively in consideration of the internal structure and characteristic of the moving picture sequence, thereby achieving a highly efficient prediction as well as encoding/decoding. In addition, the prediction system provides a sophisticated inter-video object prediction performing among two or more video objects.

Disclosure of the Invention

[0038] According to the present invention, a moving picture prediction system, for predicting a moving picture to be implemented in at least one of an encoder and a decoder, includes a plurality of memories for storing picture data for reference to be used for prediction, the plurality of memories being corresponding to different transform methods, respectively, and a prediction picture generation section for receiving a parameter representing a motion of a picture segment to be predicted, and for generating a predicted picture using the picture data stored in one of the plurality of memories used for the picture segment to be predicted based upon the parameter and one of the transform methods corresponding to the one of the plurality of memories.

[0039] The encoder generates a prediction memory indication information signal indicating the one of the plurality of memories used for generating the predicted picture and transmits the prediction memory indication information signal and the parameter to a decoding station so as to generate the predicted picture using the picture data stored in the one of the plurality of memories based upon the one of the transform methods corresponding to the one of the plurality of memories in the decoding station.

[0040] The decoder receives the parameter and a prediction memory indication information sinal indicating the one of the plurality of memories used for generating the predicted picture from an encoding station, wherein the prediction picture generation section generates the predicted picture using the picture data stored in the one of the plurality of memories based upon the parameter and the one of the transform methods corresponding to the one of the plurality of memories.

[0041] Further, according to the present invention, a moving picture prediction system, for predicting a moving picture to be implemented in at least one of an encoding and a decoding, includes a plurality of memories for storing picture data for reference to be used for prediction, the plurality of memories being assigned to different parameter effective value ranges, respectively, and a prediction picture generation section for receiving a parameter representing a motion of a picture segment to be predicted, for selecting one of the plurality of memories assigned to one of the parameter effective value ranges including a value of the parameter, and for generating a predicted picture using the picture data stored in a selected memory.

[0042] Still further, according to the present invention, a moving picture prediction system, for predicting a moving picture to be implemented in at least one of an encoding and a decoding, includes a plurality of memories for storing picture data for reference to be used for prediction and a prediction picture generation section including a motion compensator for receiving a parameter representing a motion of a picture segment to be predicted, and for generating a predicted picture by using the picture data stored in the plurality of memories based upon the parameter, and a memory update unit for updating the picture data stored in at least one of the plurality of memories at an arbitrary timing.

[0043] The moving picture prediction system predicts the moving picture in a moving picture sequence having first and second video objects, wherein the plurality of memories includes separate first and second pluralities of memories corresponding to the first and second video objects, respectively, and the prediction picture generation section includes separate first and second generators, respectively, corresponding to the first and second video objects, wherein the first generator uses the picture data stored in at least one of the first and second pluralities of memories to generate the predicted picture when predicting the first object, and generates information indicating a use of the second plurality of memories for predicting the first object, the information being added to the predicted picture.

[0044] The prediction picture generation section generates the predicted picture through a change of either one of a number and a size of the plurality of memories in response to a change in the moving picture at each time instance.

[0045] The prediction picture generation section generates the predicted picture in a limited use of memories for prediction in response to a change in the moving picture at each time instance.

[0046] The prediction picture generation section generates the predicted picture by calculating a plurality of

the predicted pictures generated by using the respective picture data stored in the plurality of memories.

[0047] The moving picture prediction system further includes a significance detector for detecting a feature parameter representing a significance of the picture segment to be predicted, wherein the prediction picture generation section generates the predicted picture by selecting at least one of choices of at least one of a plurality of prediction methods, the plurality of memories, and a plurality of memory update methods.

[0048] The moving picture prediction system further includes a significance detector for detecting a parameter representing at least one of an amount of bits available for coding the picture segment to be predicted, an amount of change of the picture segment at each time instance, and a significance of the picture segment, wherein the prediction picture generation section generates the predicted picture by selecting at least one of choices of at least one of a plurality of prediction methods, the plurality of memories, a plurality of memory update methods.

[0049] The moving picture prediction system predicts the moving picture on a video object basis, wherein the moving picture prediction system further includes a significance detector for detecting a parameter representing at least one of an amount of bits available for coding a video object to be predicted, an amount of change in the video object at each time instance, and a significance of the video object, wherein the prediction picture generation section generates the predicted picture by selecting at least one of choices of at least one of a plurality of prediction methods, the plurality of memories, and a plurality of memory update methods.

[0050] The moving picture prediction system further includes a prediction information encoder for encoding prediction relating information of the moving picture, wherein the prediction picture generation section counts times of a memory used for prediction and determines a rank of the plurality of memories based upon a counted number of the times, wherein the prediction information encoder allocates a code length to the prediction relating information to be encoded based upon the rank of a memory used for prediction.

[0051] The plurality of memories includes at least a frame memory for storing the picture data on a frame basis and a sprite memory for storing a sprite picture.

[0052] The sprite memory includes at least one of a dynamic sprite memory involving a regular updating, and a static sprite memory not involving the regular updating.

[0053] The one of the transform methods corresponding to the one of the plurality of memories is at least one of a parallel translation, an affine transformation, and a perspective transformation in an interchangeable manner.

[0054] Still further, according to the present invention, a method for predicting a moving picture to be implemented in at least one of an encoding or a decoding, includes the steps of storing picture data for reference to be used for prediction in a plurality of memories, corresponding different transform methods with the plurality of memories, respectively, receiving a parameter representing a motion of a picture segment to be predicted, and generating a predicted picture using the picture data stored in one of the plurality of memories used for predicting the picture segment based upon the parameter and one of the transform methods being corresponding to the one of the plurality of memories.

[0055] The method for predicting a moving picture further includes the steps of generating a prediction memory indication information signal indicating the one of the plurality of memories used for the picture segment to be predicted, and transmitting the prediction memory indication information signal and the parameter to a decoding station.

[0056] The method for predicting a moving picture is implemented in the decoding, and further includes the step of receiving a prediction memory indication information signal indicating the one of the plurality of memories used for generating the predicted picture and the parameter representing a motion of the picture segment to be predicted from an encoding station.

[0057] Still further, according to the present invention, a method, for predicting a moving picture to be implemented in at least one of an encoding and a decoding, includes the steps of storing picture data for reference to be used for prediction in a plurality of memories, assigning separate parameter effective value ranges to the plurality of memories, respectively, receiving a parameter representing a motion of a picture segment to be predicted, selecting one of the plurality of memories assigned to one of the parameter effective value ranges including a value of the parameter, and generating a predicted picture using the picture data stored in a selected memory.

[0058] Still further, according to the present invention, a method, for predicting a moving picture to be implemented in at least one of an encoding and a decoding, includes the steps of storing picture data for reference to be used for prediction in a plurality of memories, receiving a parameter representing a motion of a picture segment to be predicted, generating a predicted picture using the picture data stored in the plurality of memories based upon the parameter, and updating the picture data stored in at least one of the plurality of memories at an arbitrary timing.

Brief description of drawings

[0059]

Fig. 1 is a structural diagram of a moving picture encoder according to an embodiment of this invention.

Fig. 2 is a flowchart illustrating an operation of the moving picture encoder according to the embodi-

ment of this invention.

Fig. 3 is a structural diagram illustrating the configuration of a motion compensator of the moving picture encoder of the embodiment of this invention.

Fig. 4 is a flowchart illustrating an operation of the motion compensator.

Fig. 5 is a structural diagram illustrating the structure of a memory update unit of the moving picture encoder of the embodiment of this invention.

Fig. 6 is a flowchart illustrating an operation of the memory update unit.

Fig. 7 is a structural diagram illustrating the configuration of a motion compensator of a moving picture encoder according to another embodiment of this invention.

Fig. 8 is a flowchart illustrating an operation of the motion compensator of Fig. 7.

Fig. 9 is a structural diagram illustrating the configuration of a motion compensator of a moving picture encoder according to another embodiment of this invention.

Fig. 10 is a flowchart illustrating an operation of the motion compensator of Fig. 9.

Fig. 11 is a structural diagram of a moving picture encoder according to another embodiment of this invention.

Fig. 12 is a structural diagram showing the configuration of a motion compensator of the moving picture encoder according to the embodiment of this invention.

Fig. 13 is a flowchart illustrating an operation of the motion compensator of Fig. 12;

Fig. 14 is a structural diagram illustrating the configuration of a memory update unit of a moving picture encoder according to another embodiment of this invention.

Fig. 15 is a flowchart illustrating an operation of the memory update unit of Fig. 14.

Fig. 16 is a structural diagram of a moving picture encoder according to another embodiment of this invention.

Fig. 17 is a structural diagram of a moving picture encoder according to another embodiment of this invention.

Fig. 18 is a structural diagram of a moving picture encoder according to another embodiment of this invention.

Fig. 19 is a structural diagram of a moving picture encoder according to another embodiment of this invention.

Fig. 20 is a diagram showing bit stream 21 according to the first embodiment of this invention.

Fig. 21 is a diagram showing bit stream 21 according to the second embodiment of this invention.

Fig. 22 is a diagram showing bit stream 21 according to the third embodiment of this invention.

Fig. 23 is a diagram showing bit stream 21 according to the sixth embodiment of this invention.

Fig. 24 is a structural diagram of a moving picture decoder according to another embodiment of this invention.

Fig. 25 is a structural diagram illustrating the configuration of a motion compensator of the moving picture decoder according to the embodiment of this invention.

Fig. 26 is a flowchart illustrating an operation of the motion compensator.

Fig. 27 is an exemplary diagram of interpolation.

Fig. 28 is a flowchart illustrating an operation of a memory update unit of the moving picture decoder according to the embodiment of this invention.

Fig. 29 is an exemplary diagram of a video data configuration according to the VM encoding system.

Fig. 30 is an exemplary diagram of a VOP data structure.

Fig. 31 is a structural diagram illustrating the configuration of a VM encoder.

Fig. 32 is a flowchart illustrating an operation of the encoder of Fig. 31.

Fig. 33 is an exemplary diagram of VOP encoded types and corresponding prediction types.

Fig. 34 is a structural diagram illustrating the configuration of a motion compensator of the encoder of Fig. 31.

Fig. 35 is a flowchart illustrating an operation of the motion compensator of Fig. 34.

Fig. 36 is a structural diagram illustrating the configuration of a memory update unit of the encoder of Fig. 31.

Fig. 37 is a flowchart illustrating an operation of the memory update unit of Fig. 36.

Best mode for carrying out the invention

Embodiment 1.

[0060] Fig. 1 is a block diagram showing the configuration of an encoder according to a first embodiment and the following embodiments. The diagram illustrates an input moving picture signal 1, texture data 2, a motion detector 3, a motion parameter 4, a motion compensator 5, a predicted picture candidate 6, a prediction mode selector 7, a prediction mode 8, a predicted picture 9, a prediction error picture 10, a texture encoder 11, a quantized DCT coefficient 12, a locally decoded prediction error picture 13, a locally decoded picture 14, a memory update unit 15, a memory-a 16, a memory-b 17, a memory-c 18, a variable-length encoder/multiplexer 19, a transmission buffer 20, a bitstream 21, a scene-change detector 80, and a timer 81. Particularly, the motion compensator 5 and the memory update unit 15 forms a predicted picture generation section 100 which implemented a prediction system. Memories a, b, c forms a memory area 200. In the figure, portions not mentioned in this embodiment will be discussed in the

following embodiments. Fig. 2 shows a flowchart illustrating an operating flow of the encoder.

[0061] This embodiment is based upon the assumption that a plural number, e.g. three, of memories, are used adaptively according to the significance of an input moving picture based upon such characteristics of motion as an amount of motion and an intensity of color. It is also assumed that the content of an arbitrary memory (area), the memory-a for example, are updated at an arbitrary period of time and a moving picture sequence is received on a frame basis.

(1) Input signal

[0062] As aforementioned, the encoder inputs a frame representing a picture at each time instance of the moving picture sequence and decomposes the frame into the encoding units of macroblocks which are one example of picture segments subject to prediction (step S1).

(2) Adaptive use of memories

[0063] The memories store previously decoded pictures or previously provided fixed pictures. In this embodiment, the three memories are used adaptively according to the significance of a picture segment in a frame as follows.

[0064] The memory-a stores a least significant picture segment (i.e., a background-like picture segment whose motion is static or flat and the texture is flat.)

[0065] The memory-b stores a less significant picture segment (i.e., a picture segment of an object whose motion is relatively small.)

[0066] The memory-c stores a most significant picture segment (i.e., a picture segment of an object whose motion is complicated or drastic.)

[0067] The least significant picture segment to be stored in the memory-a may be a background picture segment in a video conference scene or the like. The least significant picture segment also corresponds to a background segment in a camera-work relating flat motion scene of a full screen including slightly moving objects. With this type of motion, it is efficient to obtain a frame based amount of a motion to substitute for a macroblock based motion, rather than to obtain a macroblock based amount of a motion. Specifically, a transform parameter corresponding to the sprite warping parameter discussed in the conventional art is obtained and the transform parameter of a full frame is then used as the motion parameter of a macroblock in the frame. The motion parameter may be selected from among a simple parallel translation parameter (=a motion vector), an affine motion parameter involving transformation, and a perspective motion parameter involving transformation. Here, a motion vector is one example of the motion parameter.

[0068] The less significant picture segment to be stored in the memory-b may be a picture segment of a moving figure who is not a speaker in a video conference scene or the like. This segment type of the object may be considered less attracting in the scene. The most significant picture segment to be stored in the memory-c may be a segment of an object attracting most attention in the video conference scene such as a speaker.

[0069] A picture segment stored in the memory-b or the memory-c representing a unique type of motion of an object should have a macroblock based unique motion parameter. The motion parameter of this case may be selected from among the simple parallel translation parameter (=a motion vector), the affine motion parameter involving transformation, the perspective motion parameter involving transformation, etc.

(3) Motion detection (step S2)

[0070] The motion detector 3 of this embodiment is designed to detect an arbitrary transform parameter of the respective three memories on a macroblock basis, involving no distinction between a motion vector and a warping parameter in the conventional art. The motion detector 3 is provided with additional functions of a global-motion parameter detection for detecting a frame based transform parameter using the memory-a and a local-motion parameter detection for detecting a macroblock based transform parameter using the memories a through c.

(4) Motion compensation (step S3)

[0071] Fig. 3 shows the configuration of the motion compensator 5 of this embodiment in detail. In the figure, a prediction picture memory address calculator 22, a prediction picture memory address 23, a memory reader 24, and a reference memory indicator signal 25 which is supplied externally are shown. In this embodiment, the reference memory indicator signal 25 indicates the use of the memory a, b, c. Fig. 4 shows a flowchart including steps S11 through S16 illustrating an operation of the motion compensator 5.

[0072] Initially, with an I(Intra)-frame, no motion compensation is performed (step S11). With a frame other than the I-frame, predicted picture candidates are generated based upon the global-motion and local-motion parameters corresponding to the respective memories detected in the motion detector 3 (steps S12 through S15). Specifically, the prediction picture memory address calculator 22 calculates the prediction picture memory address 23 of a predicted picture candidate in a memory identified by the reference memory indicator signal 25 based upon the motion parameter 4. Upon reception of the prediction picture memory address 23, the memory reader 24 reads out the predicted picture candidate 6 from a corresponding memory to be outputted.

[0073] In this embodiment, the global-motion and

local-motion parameters are obtained through the same transform method, thereby allowing the motion compensator 5 of Fig. 3 to be shared by both global-motion and local-motion parameter based approaches of generating a predicted picture. When generating the predicted picture candidate 6 through the global-motion parameter, the memory-a is always used as a reference memory (step S15).

(5) Prediction mode selection (step S4)

[0074] The prediction mode of this embodiment is assumed to include the following.

(a) a mode for using the memory-a,
(b) a mode for using the memory-b,
(c) a mode for using the memory-c, and
(d) a mode for using an intra-frame coding signal.

Similarly to the discussion in the conventional art, the prediction mode selector 7 selects the predicted picture candidate 6 having the least power (amplitude) of a prediction error signal, for example, from among all the predicted picture candidates 6 generated in the motion compensator 5 along with an intra-frame coding signal, and outputs a selected one of the predicted picture candidates 6 as the predicted picture 9 and an corresponding one of the prediction mode 8. The prediction mode 8 includes memory selection information indicating a memory used for predicting the selected predicted picture 9. The prediction mode 8 is transferred to the variable-length encoder/multiplexer 19 to be encoded with an allocated length of code in the bitstream 21 as prediction memory indication information 800.

(6) Memory updating

[0075] The memory update unit 15 controls the memories to be updated. Fig. 5 shows the configuration of the memory update unit 15 of this embodiment in detail. In the figure, an activity 26 used for updating the memory-a (which will be discussed later), a memory-a update judger 27, a reference memory selector 28, switches 29, 30, picture data 31 for updating the memory-a, picture data 32 for updating the memory-b, picture data 33 for updating the memory-c, and global prediction picture data 34 for updating the memory-a are shown. Fig. 6 shows a flow of a memory updating operation.

[0076] The memory updating operation of this embodiment has the following procedure. Upon reception of the locally decoded picture 14, the memory update unit 15 judges the necessity of updating the memory-a with a frame including the locally decoded picture 14 in the memory-a update judger 27 (step S17). The reference memory selector 28 selects a memory used for predicting the locally decoded picture based upon the prediction mode 8 (steps S18, S19). Then, a reference picture stored in a selected memory is updated with one of the picture data 31, 32, 33 for updating the memory-a, the memory-b, the memory-c, respectively, and the global prediction picture data 34 for updating the memory-a of the locally decoded picture 14 based upon the following rule. A memory is assumed to be updated each frame on a prediction unit (macroblock) basis.

(1) Frame based regular updating of the memory-b and the memory-c (steps S20, S21):

[0077] The locally decoded picture 14 is written into either the memory-b or the memory-c used for predicting the picture.

(2) Frame based adaptive updating of the memory-a (steps S22, S23):

[0078] The locally decoded picture 14 is written into the memory-a used for predicting the picture for an arbitrary frame only or at an arbitrary period of time based upon a memory-a update judgement 1000 obtained in step S17.

[0079] The content of memory-a is the past record of a time-unvarying picture segment such as a background picture. This removes the necessity of the regular updating of the content of memory unless a full-screen involved complicated or drastic movement such as a scene change occurs to cause a drastic change in the content of a picture segment.

[0080] As aforementioned, a frame based regular updating is performed with a complicated or drastic area of an object, whereas a longer-term based updating is performed with the content of the memory-a, thereby achieving an effective prediction with a background picture half visible among moving objects.

[0081] Viewed in this light, the memory-a is updated in an arbitrary period of time in this embodiment. Specifically, possible arbitrary update criteria are as follows.

a. A full-screen content is updated all at once with a global-motion parameter indicating a complicated or drastic motion, whereas no updating operation is performed with the parameter indicating a rather static motion.
b. A full-screen content is updated all at once on a predetermined period basis, regardless of a frame based period of time.
c. A full-screen content is updated all at once only with a frame immediately after a scene change detected.

[0082] In this embodiment, data as the arbitrary update criteria are generically called as the activity 26 used for updating memory-a. Initially, the memory-a update judger 27 judges whether to update the content of memory-a based upon the activity 26 (step S17). Specifically, the activity 26 corresponds to a value of the

global-motion parameter detected in the motion detector 3 with the arbitrary update criterion-a, a time stamp of the current frame from the timer 81 with the arbitrary update criterion-b, and a flag indicating a scene change detection outputted from the scene-change detector 80 with the arbitrary update criterion-c.

[0083] When the content of memory-a is judged to be updated, the content of the locally decoded picture 14 is outputted as the global prediction picture data 34 for updating the content of the memory-a (step S23). When no updating is judged with the content of the memory-a, then no updating is performed with the memory-a.

[0084] The memory-a update judgement 1000 of a frame is multiplexed in the bitstream 21 to be transmitted to a decoding station so that the same updating of the memory-a can be performed with the frame in the decoding station.

[0085] Fig. 20 is a diagram illustrating the bitstream 21 of this embodiment.

[0086] Fig. 20 is a conceptual diagram showing how frame data are encoded in sequence to be transmitted. Each frame data is provided with header information at the front as a frame based additional information. The header information has the memory-a update judgement 1000 multiplexed to be transmitted to a decoding station. The header information is followed by the component macroblock data of the frame. The macroblock data include the prediction memory indication information 800 indicating a memory used for predicting the macroblock data. In a counterpart decoder, a memory for predicting a predicted picture is specified based upon the prediction memory indication information 800 of macroblock data to generate the predicted picture.

[0087] Although not shown in the figures, the memory-b update information or/and the memory-c update information may be transmitted to the decoding station along with, or alternatively to, the memory-a update judgement 1000.

[0088] The aforementioned encoder thus provides the adaptive and efficient use of two or more memories in response to the content of a moving picture sequence, thereby enhancing prediction efficiency. Specifically, a moving picture sequence is predicted based upon an arbitrary transform parameter through an adaptive use of two or more memories in response to the content and characteristic of a moving picture sequence. This enables an efficient prediction of a moving picture in response to a local characteristic of a picture by even covering complicated motion. With the enhanced prediction efficiency, the encoder is allowed to reduce an amount of encoded data without deteriorating encoded picture quality. The same prediction system may be employed by the counterpart decoder for decoding a bitstream encoded through the prediction system of this invention.

[0089] This embodiment has thus disclosed the encoder performing on a frame basis. The same effect may be expected with an alternative encoder performing on an arbitrary shaped video object (VOP) basis.

[0090] Further, this embodiment has thus disclosed the encoder performing on a macroblock basis as the macroblock being a picture segment subjected to prediction. The same effect can be expected with an alternative encoder for encoding a picture on such a picture segment basis as an arbitrary shaped picture segment and a variable shaped block including fixed-size block components.

[0091] Further, this embodiment has thus disclosed the global-motion parameter detection using the memory-a. Alternatively, a single use of the local-motion parameter detection is of course applicable involving no global-motion parameter detection. With no global-motion detection, no necessity occurs for transmitting prediction information indicating a global/local prediction as the prediction mode.

[0092] Further, this embodiment may include a special memory for prediction which stores reference picture data previously generated based upon the content of the moving picture sequence. The special memory is not updated during an encoding operation.

[0093] Further, this embodiment has thus disclosed the case that memories a, b, c store a picture segment each time and the memory update unit 15 updates one of the memories a, b, c each time. If two or all of the memories a, b, c share to store a picture in part or fully, then the memory update unit 15 updates the two or all of the memories a, b, c. In the case of the memory-a being a frame memory for storing a frame of reference picture data, the memory-b being a static sprite memory involving the adaptive updating, and the memory-c being a dynamic sprite memory involving the regular updating, the memory update unit 15 does not update the memory-b as the static sprite memory for storing previously fixed reference picture data but updates the memory-a and the memory-c concurrently when the memories store the same reference picture segment. Thus, if a duplicated storage of the reference picture data occurs with the memories a, b, c, then the memory update unit 15 updates a duplicated segment stored in each memory.

[0094] The aforementioned can also be applied to the following embodiments.

[0095] Further, this embodiment has thus disclosed the use of three memories a, b, c, but alternatively, two of the memories may be utilized.

[0096] Further, a counterpart decoder may be provided with the prediction picture generation section 100 including the same components as the motion compensator 5 and the memory update unit 15 discussed in this embodiment. A motion compensator provided in the decoder, having no necessity of generating all the three predicted picture candidates, generates a single predicted picture alone based upon a decoded motion parameter.

Embodiment 2.

[0097] A second embodiment shows an encoder with a single replacement of the motion compensator 5 of the encoder shown in Fig. 1. The configuration and operation of a motion compensator 5a of the second embodiment are now described.

[0098] Fig. 7 shows the configuration of the motion compensator 5a of this embodiment in detail. The figure includes a reference memory determiner 35. Fig. 8 shows a flowchart illustrating a detailed operation of the motion compensator 5a.

[0099] Initially, with the I-frame, no compensation is performed (step S24). With a frame other than the I-frame, the reference memory determiner 35 determines a reference memory based upon a value of the motion parameter 4 (step S25). The reference memory determiner 35 holds effective motion parameter value ranges (which will be discussed later in detail) allocated, respectively, to the memories a, b, c. The reference memory determiner 35 compares the respective effective motion parameter value ranges with the value of the motion parameter 4 to judge which memory is designated by the motion parameter 4 and outputs a reference memory indicator signal 25a for identifying the respective memories a, b, c.

[0100] The effective motion parameter value ranges are effective search ranges allocated to the respective memories for detecting a motion vector, for example. Specifically, if ±15 pixels are assumed to be given for a total search value range, then the memory-a is chosen to be used for prediction in a range of ±0 to 3 pixels, the memory-b is used in a range of ±4 to 8 pixels, and the memory-c is used in a range of ±9 to 15 pixels, for example. Here, the reference memory determiner 35 operates only when the local-motion parameter is used for prediction, because the memory-a is exclusively used as a reference memory when the predicted picture candidate is generated based upon the global-motion parameter. This motion-vector value based approach of identifying a memory to be used for prediction is based upon the assumption that a background picture should include a static motion and a most attractive picture should include a complicated or drastic motion. This motion-vector value based approach of identifying a memory for prediction involves no necessity of encoding the prediction mode to be transmitted.

[0101] Next, the predicted picture candidate 6 is generated based upon the reference memory indicator signal 25a of a selected memory (steps S26 through S30). Specifically, the prediction picture memory address calculator 22 calculates the prediction picture memory address 23 of the predicted picture candidate 6 in an identified memory by the reference memory indicator signal 25a based upon the motion parameter 4. Based upon the prediction picture memory address 23, the memory reader 24 reads out the predicted picture candidate 6 from the memory to be outputted.

[0102] As the global-motion and local-motion parameters of this embodiment are based upon the same transform method, both parameter based approaches can share the motion compensator 5a of Fig. 7 for generating the predicted picture candidate. When reading out the predicted picture candidate 6 based upon the global-motion parameter (step S31), the memory-a is always used as a reference memory.

[0103] The effective motion parameter value ranges may be fixed on a moving picture sequence basis, and alternatively, changed on a frame basis, for example. With the frame based changes, the effective motion parameter value ranges assigned to the respective memories of the frame are multiplexed in a bitstream to be transmitted to a decoding station to perform the same memory selection.

[0104] Fig. 21 is a diagram showing the bitstream 21 of this embodiment.

[0105] The bitstream is provided with header information added at the front on a moving picture sequence basis. The header information includes effective motion parameter value range indication information of the respective memories. By thus designating the effective motion parameter value range indication information at the front of a moving picture sequence, the moving picture sequence is predicted with the fixed effective motion parameter value ranges in a counterpart decoder.

[0106] When varying the effective motion parameter value ranges each frame, the effective motion parameter indication information is to be included in the header information added on a frame basis.

[0107] Thus, the efficient and adaptive use of the memories in response to the magnitude of a local motion of a frame can be provided by the encoder including the motion compensator 5a, thereby enhancing prediction efficiency.

[0108] This embodiment has thus disclosed the encoder performing on a frame basis. The same effect may be expected with an alternative encoder performing on an arbitrary shaped video object (VOP) basis.

[0109] Further, this embodiment has thus disclosed the encoder performing on a macroblock basis. The same effect may be expected with an alternative encoder for encoding a picture on such a picture segment basis as an arbitrary shaped picture segment and a variable shaped block including fixed-size block components.

[0110] Further, this embodiment has thus disclosed the global-motion parameter detection using the memory-a. Alternatively, a single use of the local-motion parameter detection is of course applicable involving no global-motion parameter detection. No global-motion detection involves no necessity for transmitting information indicating a global/local prediction as the prediction mode.

[0111] Further, a counterpart decoder may be provided with the prediction picture generation section 100

including the same component as the motion compensator 5 discussed in this embodiment. In the decoder, a motion compensator only generates a single predicted picture based upon a decoded motion parameter.

Embodiment 3.

[0112] Another embodiment shows an encoder having a single replacement of the motion compensator 5 of the encoder of Fig. 1. The configuration and operation of a motion compensator 5b are now described. A motion detector 3a employed in this embodiment is assumed to output an amount of the parallel translation, the affine parameter, and the perspective parameter as motion parameters 4a.

[0113] Further, the memory-a of this embodiment is assumed to be a frame memory for storing a reference picture frame, the memory-b is assumed to be a static sprite memory, and the memory-c is assumed to be a dynamic sprite memory.

[0114] Fig. 9 shows the configuration of the motion compensator 5b of this embodiment in detail. In the figure, a parallel translation amount 36 (i.e., a motion vector), an affine parameter 37, a perspective parameter 38, a parallel-translation based prediction picture memory address calculator 39, an affine parameter based prediction picture memory address calculator 40, and a perspective parameter based prediction picture memory address calculator 41 are shown. Fig. 10 is a flowchart illustrating the operation of the motion compensator 5b in detail.

[0115] Initially, with the I-frame, no prediction is performed (step S33). With a frame other than the I-frame, the prediction picture memory address calculators 39 through 41 calculate the respective prediction picture memory addresses 23 based upon the respective values of the motion parameters 4a (step S34).

[0116] The memory address calculators 39, 40, 41 calculate addresses based upon picture transform methods assigned, respectively, to the corresponding memories. In this embodiment, the parallel translation is assigned to the memory-a, the affine parameter involving such a simple transformation as a rotation and an expansion/contraction is assigned to the memory-b, and the perspective parameter involving a three-dimensional complicated motion is assigned to the memory-c. These transform methods may be expressed by the following transform expressions.

[Parallel translation]

[0117] Amount of parallel translation (a, b):

$$x'=x+a$$

$$y'=y+b$$

(Affine transform]

[0118] Affine parameter (a,b,c,θ):

$$x'=a(\cos\theta)x+a(\sin\theta)y+b$$

$$y'=a(-\sin\theta)x+a(\cos\theta)y+c$$

[Perspective transform]

[0119] Perspective parameter (a,b,c,d,e,f):

$$x'=(ax+by+c)/(gx+hy+1)$$

$$y'=(dx+ey+f)/(gx+hy+1)$$

[0120] Here, (x, y) in a two-dimensional coordinate system represents a pixel location of an original macroblock. (x', y') represents a pixel location in a memory corresponding to (x, y,) based upon each of the parameters. That is a location in a memory (x', y') is calculated based upon these parameters. Through this mechanism, a memory most suitable for the characteristic of a motion can be chosen to be used for prediction on a macroblock basis. With calculated prediction picture memory addresses 23 based upon the respective motion parameters 36, 37, 38, the memory reader 24 reads out the predicted picture candidates 6 from corresponding memories to be outputted (steps S35 through S39).

[0121] The transform methods assigned to the respective memories of the frame are multiplexed in the bitstream 21 to be transmitted to a decoding station as a motion detection method indication information so that the same motion compensation can be performed in the decoding station.

[0122] Fig. 22 is a diagram showing the bitstream 21 of this embodiment.

[0123] Header information added at the front of a moving picture sequence includes the motion detection method indication information. In the encoder, the transformation types to be used in the respective memories are interchangeable, and thus the motion detection method indication information indicating a memory-transform method relation is to be transmitted to the counterpart decoder as the header information of the moving picture sequence. Thus, transformation types assigned to be used with the respective memories can be identified in the decoder.

[0124] In the decoder, the identified transformation types are dynamically assigned to the respective memories.

[0125] Thus, the efficient and adaptive use of the memories in response to the characteristic of a local motion of a frame is provided by the encoder including the motion compensator 5b, thereby enhancing prediction efficiency.

[0126] This embodiment has thus disclosed the

encoder performing on a frame basis. The same effect may be expected with an alternative encoder performing on an arbitrary shaped video object (VOP) basis.

[0127] Further, this embodiment has thus disclosed the encoder performing on a macroblock basis. The same effect may be expected with an alternative encoder for encoding a picture on such a picture segment basis as an arbitrary shaped picture segment and a variable shaped block including fixed-size block components.

[0128] Further, this embodiment has thus disclosed the global-motion parameter detection using the memory-a. Alternatively, a single use of the local-motion parameter detection is of course applicable involving no global-motion parameter detection. No global-motion detection involves no necessity of transmitting information of a global/local prediction as the prediction mode.

[0129] Further, this embodiment has thus disclosed the use of the memories a, b, and c. Alternatively, the use of memories a and b alone, memories a and c alone, or memories b and c alone, is also applicable.

[0130] Further, a decoder may be provided with the prediction picture generating section 100 including the same component as the motion compensator 5b discussed in this embodiment. A motion compensator in a decoder only generates a single predicted picture based upon a decoded motion parameter.

Embodiment 4.

[0131] Another embodiment shows an encoder which receives a plural number, two for example, of different video objects, having shape information, intermingled in a moving picture sequence for a collective encoding. Fig. 11 shows the configuration of the encoder of this embodiment.

[0132] In the figure, an input picture frame 42, an object separator 43, object data 44a, 44b, shape blocks 45a, 45b, switches 46a, 46b, shape encoders 47a, 47b, compressed shape block data 48a, 48b, locally decoded shape blocks 49a, 49b, texture data (macroblocks) 50a,50b, motion detectors 51a, 51b, motion parameters 52a, 52b, motion compensators 53a, 53b, predicted picture candidates 54a, 54b, prediction mode selectors 55a, 55b, prediction mode information 56a, 56b, predicted pictures 57a, 57b, prediction error signals 58a, 58b, texture encoders 59a, 59b, compressed texture data 60a, 60b, locally decoded prediction error signals 61a, 61b, locally decoded macroblocks 62a, 62b, a memory update unit 63, memory-a 64, memory-b 65, memory-c 66, memory-d 67, memory-e 68, memory-f 69, variable-length encoders 70a, 70b, a multiplexer 71, a buffer 72, a bitstream 73, a memory section 94, an object-A encoder 88a for encoding an object-A, and an object-B encoder 88b for encoding an object-B are shown. The object encoders 88a, 88b are structurally identical to each other with the identical components.

[0133] This encoder inputs the picture frame 42, which is decomposed into the encoding units of objects in the object separator 43. The object separator 43 is assumed to be assigned a processing method arbitrarily.

[0134] The shape information of an object is transferred to the shape encoder 47a, 47b in a form of the shape block 45a, 45b to be encoded, and then transferred to the variable-length encoder 70a, 70b as the compressed shape block data 48a, 48b.

[0135] The motion detector 51a, 51b detects a motion parameter based upon the locally decoded shape block 49a, 49b in the same manner as that of the VM encoding system. A motion parameter can be detected on a macroblock basis by using all the memories a through f.

[0136] As a rule, however, the memories a through c are designed to be used for an object-A to be encoded in the object-A encoder 88a, and the memories d through f are designed to be used for an object-B to be encoded in the object-B encoder 88b.

[0137] Also, as for a motion type, an arbitrary transform parameter is assumed to be detected on a macroblock basis with all the memories in the memory section 94, involving no distinction between the motion vector and the warping parameter.

[0138] The motion compensator 53a, 53b generates all the predicted picture candidates 54a, 54b based upon the respective motion parameters 52a, 52b. Then, in the prediction mode selector 55a, 55b, the predicted picture 57a, 57b is obtained along with the prediction mode information 56a, 56b. The predicted picture 57a, 57b is then differentiated from an original signal or the texture data 50a, 50b to obtain the prediction error signal 58a, 58b, which is encoded in the texture encoder 59a, 59b to be transmitted to the variable-length encoder 70a, 70b. The locally decoded prediction error signal 61a, 61b is added to the predicted picture 57a, 57b to obtain the locally decoded macroblock 62a, 62b to be stored into the memories a through f in accordance with an indication by the memory update unit.

[0139] Object A/B data when encoded in the object-A/B encoder 88a, 88b are multiplexed in the bitstream 73 at the multiplexer 71 to be transmitted via the buffer 72.

[0140] The prediction of this embodiment is discussed below focusing on the motion compensator 53a, 53b playing a primary role in the prediction.

[0141] The motion compensator 53a, 53b of this embodiment generates a predicted picture candidate based upon the motion parameter 52a, 52b detected in the motion detector 51a, 51b. Fig. 12 shows the configuration of the motion compensator 53a in detail. Fig. 13 shows a flowchart illustrating the operation of the motion compensator 53a in the object-A encoder 88a.

[0142] In Fig. 12, an object-B reference judger 74a and an object-B reference indicator flag 75a are shown.

[0143] The motion parameter 52a includes memory information used for detection. A predicted picture can-

didate is generated based upon a parameter value through the prediction picture memory address calculator 22a and a memory reader 24a in the same manner as that stated in the first embodiment (step S44 through step S49). The object-B reference judger 74a judges if the memories assigned to object-B are used for predicting the current macroblock based upon the reference memory information included in the motion parameter 52a (step S43).

[0144] The object-B reference judger 74a outputs a judged result as the object-B reference indicator flag 75a, which is multiplexed in the bitstream 73 to be transmitted to a decoding station so as to be used for deciding whether the object can be reproduced in a single use of the memories a, b, c of self in the decoding station. In order to secure the single use of the memories of self when reproducing the object in the decoding station, a limited use of the memories (a, b, c alone) of self for prediction can be controlled by an externally supplied signal 85a at the time of encoding the object.

[0145] Thus, the efficient and adaptive use of the memories in response to the characteristic of a local motion of a frame is provided by the encoder including the motion compensator 53a, 53b, thereby achieving an efficient prediction.

[0146] This embodiment has thus disclosed the encoder for encoding an object on a macroblock basis. The same effect may be expected with an alternative encoder for encoding a picture on such a picture segment basis as an arbitrary shaped picture segment and a variable shaped block including fixed-size block components.

[0147] Further, a decoder may be provided with the same components as the motion compensator 53a, 53b of this embodiment. A motion compensator 53 of the decoder only generates a single predicted picture based upon a decoded motion parameter. Further, if the decoder is structured so as to acknowledge whether a decoding object can be reproduced by itself by way of decoding a bit corresponding to object reference indicator flag 75a, 75b of the other object in a bitstream, then an error-free secured reproduction of decoded object data can be achieved.

Embodiment 5.

[0148] Another embodiment shows an encoder where the number of memories or the size of a memory can be varied flexibly in response to a change in a video object at each time instance. The encoder of a fifth embodiment modifies the encoder of Fig. 1 with a replacement of the memory update unit 15.

[0149] Fig. 14 shows the configuration of a memory update unit 15a of this embodiment in detail. In the figure, a memory expansion judger 76, a memory expansion indicator signal 77, and a memory contraction judger 78, and a memory contraction indicator signal 79 are shown. Fig. 15 shows an operating flow (step S51 through S63) of the memory update unit 15a.

[0150] A picture substantially different from the past record of a moving picture sequence stored in the memories may occur due to a scene change or the like. This may cause a deterioration of prediction efficiency after the scene change if reference pictures stored in the existing memories are the only available. For such an occasion, the scene-change detector 80 detects a scene change, a frame appearing immediately after the detected scene change is subject to intra-frame coding or the like, and resultant intra-frame coded data are stored additionally in a memory as new reference data, thereby enhancing prediction efficiency thereafter.

[0151] Further, in consideration of the physical limitation of storage capacity, a flexible approach of contracting the portions which are rarely used for prediction of the reference pictures stored in the memories is introduced. Specifically, the frequency in use of memory areas for prediction of the respective memories a, b, c is examined in the memory update unit 15a based upon the prediction mode 8. Consequently, the memory update unit releases a memory area identified low in frequency from an area for use. For example, with a software based implementation of this encoder, limited RAM resources may be used effectively.

[0152] Viewed in this light, the memory update unit 15a of this embodiment is provided with a function of expanding a memory area in response to each time instance of a time-varying moving picture sequence and contracting a memory area including a reference picture rarely used for prediction.

[0153] The memory-a, similarly to the first embodiment, is judged in the memory-a update judger 27 whether to be updated (step S50). When updating the memory-a, the locally decoded picture 14 is written into the memory-a (steps S56, S57). The locally decoded picture 14 is written into the other memories as well in accordance with the prediction mode 8 (step S51 through S55).

[0154] The updating of the contents of the memories involves the judgement of memory expansion/contraction. The memory expansion judger 76 judges whether to expand the size of the memory-a (or the memory-b, or the memory-c) based upon the activity 26 used for updating the memory-a (steps S58 through S60). When a positive judgement is made due to a scene change or the like, the expansion of the memory is indicated by the memory expansion indicator signal 77. The memory contraction judger 78 counts the times of a memory area used for prediction based upon the prediction mode 8. With a memory area counted less than a predetermined number in use for prediction, the contraction of the memory area is indicated by the memory contraction indicator signal 79 (steps S61 through S63).

[0155] Thus, a highly efficient prediction can be achieved in response to each time instance of a time-varying moving picture sequence by the encoder including the memory update unit 15a. In addition, the

dynamic allocation of memory areas required for prediction contributes to the enhancement of prediction efficiency and the effective use of memory resources.

[0156] This embodiment has thus disclosed the encoder performing on a frame basis. The same effect may be expected with an alternative encoder performing on an arbitrary shaped video object (VOP) basis.

[0157] Further, this embodiment has thus disclosed the encoder for encoding a frame on a macroblock basis. The same effect can be expected with an alternative encoder for encoding a picture on such a picture segment basis as an arbitrary shaped picture segment and a variable shaped block including fixed-size block components.

[0158] Further, a counterpart decoder may be provided with the same component as the memory update unit 15a discussed in this embodiment.

Embodiment 6.

[0159] With reference to the respective previous embodiments, the memories to be used for prediction are changed on a macroblock basis. Alternatively, the memories to be used for prediction can be changed on a frame or a video object basis. This eliminates the necessity of encoding memory relating information to be encoded on a frame or a video object basis and memory selection information (which is included in the prediction mode 8) to be encoded on a macroblock basis, thereby achieving an efficient encoding.

[0160] With reference to the encoder of Fig. 1 of the first embodiment, for example, the macroblock based changes of the memories used for prediction create the necessity of transmitting additional information identifying a memory used for prediction on a macroblock basis. According to this embodiment, the changing unit of the memories to be used for prediction is limited to a frame or a video object, thereby eliminating the additional information to be transmitted on a macroblock basis effectively. Fig. 23 shows a difference of the transmission bitstream 21 of this embodiment from the transmission bitstream 21 of Fig. 20 of the first embodiment. The bitstream of Fig. 23 represents a frame based change of the memories to be used for prediction with the prediction memory indication information 800 included in frame based header information. The bitstream of Fig. 23 may be effective, for example, in the case that the picture characteristic of a moving picture sequence changes infrequently including little changes locally on a macroblock level. Further, a decoder may be provided so as to decode the thus encoded bitstream to reproduce a frame or a video object.

Embodiment 7.

[0161] With reference to the previous embodiments, two predicted picture candidates read out from an arbitrary plural number, two (e.g., memories a and b) for example, of the memories are subject to an arithmetic mean to obtain a picture as a member of the predicted picture candidates 6 or as the predicted picture 9. Further, a decoder may be provided so as to decode the thus encoded bitstream to reproduce a frame or a video object.

Embodiment 8.

[0162] With reference to the encoders of the previous embodiments, a previously detected feature parameter representing the spatial complexity, perceptual significance and the like of a picture segment as a prediction unit may be utilized as tools for deciding a prediction mode and for judging the updating of the memories.

[0163] For example, a moving picture is assumed to include a motion too complicated to encode data in an acceptable quality within a given amount of encoding. In this case, significance is examined on a prediction picture segment (e.g., a macroblock, an arbitrary shaped picture segment, an arbitrary shaped block) basis. Consequently, a low quality encoding is assigned to some extent to a less significant segment in order to save some amount of encoding for a more significant segment, thereby improving an overall picture quality. With the encoders of this invention where two or more memories are switched interchangeably at an arbitrary timing to be used for prediction, a more adaptive prediction can be achieved in response to the characteristic of a picture, through detecting a feature parameter representing the significance of a prediction picture segment and then determining the use of the memories dynamically based upon a detected feature parameter. For example, as shown in Fig. 16, a segment-significance detector 95 is provided for detecting the feature parameter on a segment basis to determine the significance of the segment. The segment-significance detector 95 transfers a segment-significance to a prediction mode selector 7a and a quantization parameter based upon the segment-significance to a texture encoder 11a. With a segment judged more significant in the segment-significance detector 95, a most complex mode among two or more prediction modes available is used for prediction. Specifically, reference pictures from the respective memories a, b, c are used to obtain the motion parameters and the predicted pictures, respectively, based upon a complex motion model. In the prediction mode selector 7a, a prediction mode having the highest prediction efficiency is selected from among modes including an arbitrary combination (e.g., an arithmetic mean) of the predicted pictures. At the same time, reference pictures of all the memories used for prediction are updated. The texture encoder 11a performs an encoding using a quantization parameter having a smaller quantization step size. With a less significant segment, a simplified prediction mode (i.e., a parallel translation amount detection using a single memory) is employed for prediction and a quantization parameter having a larger

quantization step size is utilized for encoding, regardless of the amplitude of an obtained prediction error signal, so that an amount of encoding be reduced. Through this control, a less significant segment reduces its picture quality to some extent and a more significant segment maintains its quality through a high-quality prediction, thereby improving an overall quality within a given amount of encoding.

Embodiment 9.

[0164] In an encoder where a moving picture sequence is predicted and encoded by using two or more memories, a parameter representing an amount of encoding available for the moving picture sequence at each time instance, an amount of a change in a scene at a certain time instance (e.g., a scene change detection), or the feature parameter or significance of a prediction picture segment described in the eighth embodiment may be detected previously. The values of these parameters may be used for predicting a picture at a particular time instance in a prediction system. Alternatively, these values may be utilized as judging tools for selecting a reference memory area in a prediction system. A frame-significance detector 96 may be provided for determining the significance on a frame basis as shown in Fig. 17. The frame-significance detector 96 detects, for example, an amount of a change in a motion between the current and the previous frame (e.g., a scene change detection by the scene change detector 80), the appearance of a new object or the disappearance of an object or the like. A final significance of the current frame is determined in consideration of an amount of encoding available for the current frame informed by the transmission buffer 20. Based upon the final significance, a more significant frame may be predicted by using all the prediction methods and reference memory areas available for the maximum possible improvement of prediction efficiency, whereas a less significant frame is predicted in a limited use of the prediction methods and the reference memory areas for a simplified encoding so as to reduce the throughput. An alternative encoder for performing intra-frame coding alone involving no prediction at a scene change may be possible. In addition, a more sophisticated quality control may be achieved with a joint use of the segment-significance detector 95 discussed in the eighth embodiment. Through this control, a less significant frame reduces its quality to some extent and a more significant frame maintains its quality by a high-quality prediction, thereby improving an overall picture quality within a given amount of encoding.

[0165] The idea of this embodiment is also applicable to a software based encoding being associated with unsteady transaction processes and unsteady size of available storage in order to achieve an efficient encoding in the maximum use of available resources. This reduces a throughput with a less significant frame, thereby accelerating a general processing speed.

Embodiment 10.

[0166] In an encoder where a moving picture sequence including two or more video objects is predicted and encoded by using two or more memories, as shown in Fig. 11, a parameter representing a gross amount of encoding available for the sequence, an available amount of encoding of a video object at each time instance, an amount of a change in a video object at each particular time instance (e.g., the appearance/disappearance of an object), a level of significance/attention of a video object in a particular scene, or the feature parameter or significance of a prediction picture segment discussed in the eighth and ninth embodiments may be detected previously. The values of these parameters may be utilized for predicting a video object at each particular time instance. Alternatively, these values may be utilized as judging tools for selecting a reference memory area.

[0167] For example, as shown in Fig. 18, significance detectors 97a through 97n responsive, respectively, to objects 1 through n may be provided for detecting a parameter representing an amount of a change in an object at each time instance, or the appearance/disappearance of an object. In addition, the significance of an object at each time instance is determined in consideration of the occupational proportion of a buffer 72x for storing the encoded data of all the objects and the occupational proportion of virtual buffers 72a through 72n for the respective objects. When a new type of segment appears as a result of an object having another object overlapped in part, for example, then the following control may be applied to this type of segment because this type of segment has a great influence on prediction efficiency thereafter. A higher significance may be assigned to this type of segment to obtain an encoded picture in high quality even without enough space available for storage in the corresponding virtual buffer of the object. Significance detected in the significance detectors 97a through 97n is transferred to object 1 through N encoders 98a through 98n, where a full use of the prediction methods and the reference memory areas available is allowed to a more significant object so as to improve prediction efficiency to the maximum, whereas a limited use of the prediction methods and the reference memory areas is assigned to a less significant object so as to simplify the encoding, thereby reducing the throughput. Further, with an encoder for encoding objects decomposed from a frame through a real time separation, when a considerable amount of a change occurs in the contents of the object due to the appearance of a new object or the disappearance of an existing object, the object may be subject to intra-frame coding alone with no prediction involved. A more sophisticated quality control may be achieved on a prediction segment basis of an object in a joint use of the object 1

through N encoders 98a through 98n and the segment-significance detector 95 discussed in the eighth embodiment. Through this control, a less significant object is reduced in quality to some extent and a more significant object manages to maintain its quality through a sophisticated prediction, thereby improving an overall quality within a given amount of encoding.

Embodiment 11.

[0168] An alternative encoder may be provided with a prediction information encoder 91 for allocating a code (encoding) to prediction relating encoding information (e.g., a reference memory number) as shown in Fig, 19.

[0169] In the encoder where a moving picture sequence or a video object is predicted and encoded by using the memories a, b, c, the memories may be ranked based upon frequency in use for prediction with ranks being updated dynamically during an encoding operation. Consequently, a code allocation is performed to the prediction relating encoding information (e.g., a reference memory number) based upon the ranks of the respective memories used for prediction in the prediction information encoder 91.

[0170] For example, in the encoder of Fig. 19, the memory update unit 15b may be provided with a counter 92 which counts the times of the respective memories a, b, c to be used for prediction, ranks the memories a, b, c based upon counted values, and outputs resultant ranking information 90. This ranking may be performed on a picture (VOP) basis at a particular time instance of a frame or a video object, and alternatively, on a smaller unit basis of a prediction picture segment (e.g., a macroblcok, an arbitrary shaped segment, and an arbitrary shaped block).

[0171] This shows how often the respective memories are used for prediction. A memory in a frequent use for prediction is the most significant for prediction, and thus, a high frequency in use for reference corresponding to a high rank.

[0172] When encoding information on the frequency in use for prediction of the memories on a prediction picture segment basis, a memory in a frequent use for reference (i.e., a high-rank memory) is allocated a short code to enhance encoding efficiency.

[0173] In addition, if the motion parameter detected on a prediction picture segment basis is allocated a code length in response to the rank of a memory used for reference, then a shorter code may be assigned to a motion parameter value generated frequently, thereby achieving an efficient encoding of the prediction information. This may be materialized with an alternative encoder where the prediction information encoder 91 in the variable-length encoder/multiplexer 19 receives the ranks of the respective memories from the counter 92 in the memory update unit 15b and encodes the prediction information using a variable-length code based upon the ranking information 90.

Embodiment 12.

[0174] Fig. 24 shows the configuration of a picture decoder where an encoded digital picture through compression is reproduced through expansion according to another embodiment. In the figure, the encoded bitstream 21, a variable-length decoder 119, the quantized DCT coefficient 12, a quantization orthogonal transform coefficient 12a, a quantization step 12b, a texture decoder 111, a dequantizer 111a, an inverse orthogonal transformer 111b, a decoding adder 190, a decoded picture 101, a display controller 191, the prediction mode 8, memory-b update information 1001, memory-c update information 1002, the motion vector 4 (a motion parameter), the prediction memory indication information 800, an in-screen location 195 of a prediction picture segment, a motion compensator 105, a memory-a 116, a memory-b 117, a memory-c 118, a memory update unit 115, and a predicted picture 106 are shown. The motion compensator 105 and the memory update unit 115 form a prediction picture generation section 100a. The memories a, b, c form a memory area 200a.

[0175] According to this embodiment, the memory-a is assumed to be a frame memory designed to store a frame of picture data, the memory-b is assumed to be a static sprite memory, and the memory-c is assumed to be a dynamic sprite memory. The decoder of this embodiment is assumed to receive the bitstream 21 of Fig. 22. Although not shown in Fig. 22, the memory-b update information 1001 and the memory-c update information 1002 are assumed to be transmitted in the bitstream. The memory-b update information 1001 is assumed to include an update indication for a full updating of the static sprite memory and picture data for the full updating. Similarly, the memory-c update information 1002 is assumed to include an update indication for a full updating of the dynamic sprite memory and picture data for the full updating.

[0176] The operation of the thus configured decoder is described below. The variable-length decoder 119 analyzes the bitstream 21 and decomposes it into separate encoded data. The quantization orthogonal transform coefficient 12a is transferred to the dequantizer 119a to be dequantized by using the quantization step 12b. A dequantized result is subject to inverse orthogonal transformation in the inverse orthogonal transformer 111b to obtain a decoded texture, which is transferred to the decoding adder 190. Orthogonal transformation employed here is the same as that employed in an encoding station such as Discrete Cosine Transformation (DCT).

[0177] The motion compensator 105 inputs the motion vector 4, the prediction memory indication information 800, and information indicating the in-screen location 195 of a prediction picture segment included in the bitstream 21, all of which are decoded in the variable-length decoder 119. The motion compensator 105 reads out a right predicted picture from reference pic-

tures stored in the memories a, b, c based upon the three kinds of information. The in-screen location 195 of a prediction picture segment can be obtained by counting the number of macroblocks, other than from the information included in the bitstream. The process of generating a predicted picture will be discussed in a later section for describing the operation of the motion compensator 105 in detail.

[0178] The decoding adder 190, based upon the information of the prediction mode 8, outputs an output from the inverse orthogonal transformer 111b directly as the decoded picture 101 with a block through intra-frame coding, and, with a block through inter-frame coding, adds an output from the inverse orthogonal transformer 111b to the predicted picture 106 to be outputted as the decoded picture 101. The decoded picture 101 is transferred to the display controller 191 to be outputted to a display device and also transferred to the memories a, b, c to be stored as a reference picture for a later use in decoding. A memory writing operation is controlled by the memory update unit 115 based upon the prediction mode 8.

[0179] A predicted picture generation performed in the motion compensator 105 in the prediction picture generation section 100a is now discussed. According to this embodiment, the prediction method of a picture is determined based upon the prediction memory indication information 800. The decoder of this embodiment generates a predicted picture using a reference picture through predetermined coordinate transformation and interpolation based upon the motion vector 4 and the prediction memory indication information 800. Coordinate transform methods are assigned previously to the respective memories to be used for prediction. For example, the following approaches are possible similar to the picture transform methods described in the third embodiment.

(1) the memory-a used for prediction (with the prediction memory identification information 800 indicating the use of the memory-a)

[0180] The coordinates of each pixel of a prediction segment are translated based upon the motion vector and picture data at a corresponding location in the memory-a is read out as the predicted picture.

(2) the memory-b used for prediction (with the prediction memory identification information 800 indicating the use of the memory-b)

[0181] An affine transform expression is found based upon the motion vector, the coordinates of each pixel of a prediction segment are displaced based upon the transform expression, and picture data at a corresponding location in the memory-c is read out as a predicted picture.

(3) the memory-c used for prediction (with the prediction memory identification information 800 indicating the use of the memory-c)

[0182] A perspective transform expression is found based upon the motion vector, the coordinates of each pixel of a prediction segment are displaced based upon the transform expression, and picture data at a corresponding location in the memory-b is read out as a predicted picture.

[0183] Fig. 25 shows the configuration of the motion compensator 105 in detail. In the figure, a switch 161, a corresponding point determiner 162 for the memory-a, a corresponding point determiner 163 for the memory-b, a corresponding point determiner 164 for the memory-c, a memory read address generator 165, a switch 166, and an interpolator 167 are shown. Fig. 26 is a flowchart illustrating the optation of the motion compensator 105.

[0184] The operation of the motion compensator 105 of this embodiment is described below with reference to Figs. 25 and 26.

1) Determining a corresponding point

[0185] Initially, the corresponding point determiner of a corresponding memory is selected by the switch 161 based upon the prediction memory indication information 800. The vector 4 is then inputted to a selected corresponding point determiner. In this section, a predicted picture location corresponding to each memory is calculated, which is explained below with each memory.

1-1) the memory-a indicated by the prediction memory indication information 800 (step S100)

[0186] A predicted picture location is calculated through parallel translation based upon a motion vector (step S101). Specifically, a predicted picture location (x',y') corresponding to a pixel at a prediction picture segment location (x,y) is determined based upon a motion vector (a, b) according to the following expression.

$$x' = x + a$$

$$y' = y + b$$

A determined predicted picture location is outputted to the memory read address generator 165.

1-2) the memory-b indicated by the prediction memory indication information 800 (step S103)

[0187] An affine transform expression is determined based upon the motion vector 4. Specifically, an affine parameter (a, b, c, θ) of the following expression is determined by using the motion vector of a vertex of a rectangular area enclosing a prediction picture seg-

ment,

$$x' = a(\cos\theta)x + a(\sin\theta)y + b$$

$$y' = a(-\sin\theta)x + a(\cos\theta)y + c$$

thereby obtaining the predicted picture location (x',y') corresponding to a pixel at the location (x,y) of a prediction picture segment to be outputted to the memory read address generator 165 (step S104).

1-3) the memory-c indicated by the prediction memory indication information 800 (step S106)

[0188] A perspective transform expression is determined based upon a motion vector. Specifically, a perspective parameter (a, b, c, d, e, f) of the following expression is determined by using the motion vector of a vertex of a rectangular area enclosing a prediction picture segment,

$$x' = (ax + by + c) / (gx + hy + 1)$$

$$y' = (dx + ey + f) / (gx + hy + 1)$$

thereby obtaining the predicted picture location (x',y') corresponding to a pixel at the location (x,y) of a prediction picture segment to be outputted to the memory read address generator (step S107).

2) Reading out data for generating a predicted picture

[0189] Based upon the predicted picture location (x', y') outputted from a selected corresponding point determiner, the memory read address generator 165 generates a memory address for specifying the location of picture data required for generating a predicted picture in a reference picture stored in a memory, and reads out the data for generating a predicted picture (steps S102, 105, 108).

3) Generating a predicted picture

[0190] Among the component pixels of a predicted picture, with a pixel at an integer pixel location, the data for generating a predicted picture is used directly as a component pixel of a predicted picture, and with a pixel at a real number precision pixel location, the data for generating a predicted picture is subject to interpolation in the interpolator 167 to generate an interpolated pixel value (steps S109, S110, S111). Fig. 26 illustrates an interpolated pixel value generation. In Fig. 26, (i_0, j_p) denotes an integer pixel location, (j^p, j^p) denotes a real number precision pixel location, and w denotes a weight.

4) Updating a memory (a reference picture)

[0191] Fig. 28 shows a flowchart illustrating a control operation of the memory update unit 115. The memory update unit 115 controls an updating of the respective memories on a readout unit basis (e.g., a macroblock) of a predicted picture based upon the prediction mode 8 (or the prediction memory indication information 800). With the memory-a used for prediction (step S112), the contents of the memory-a and the memory-c are updated regularly with the decoded picture 101 (step S113). With the memory-b used for prediction (step S114), the reference picture of the memory-b is not updated on a readout unit basis of a predicted picture because of the memory-b being a static sprite memory, whereas the contents of the memory-a and the memory-c are updated regularly with the decoded picture 101 (step S115). When receiving the update indication by the memory-b update information 1001, then the memory update unit updates a full content of the memory-b with received picture data included in the memory-b update information 1001 (step S116). With the memory-c used for prediction (step S117), the contents of the memory-a and the memory-c are updated regularly by using the decoded picture 101 (step S118). When receiving the update indication by the memory update information, the memory update unit updates the content of the memory-c with received picture data included in the memory-c update information 1002 (step S119).

[0192] The use of the three memories a, b, c of this embodiment may be replaced by the use of two memories thereof, for example, with the memories a and b, that is, a frame memory and a static sprite memory. Alternatively, the memories a and c, that is, a frame memory and a dynamic memory, may be used.

[0193] As aforementioned, according to the decoder of this embodiment, the bitstream 21 encoded through an efficient prediction using the various kinds of motion parameters in response to the motion of a picture can be decoded. In addition, the decoder is applicable to the arbitrary updating approach of the contents of a reference picture at a timing determined in the encoding station, thereby achieving a more adaptive decoding in response to the characteristic of a picture.

[0194] According to this embodiment, if the bitstream includes a prediction error signal encoded through encoding other than orthogonal transform encoding, the same effect may be obtained by replacing a component for decoding a prediction error signal, other than the motion compensator and memory update unit.

[0195] Further, this embodiment may be applied not only to a decoder for decoding data on a fixed-size block basis, e.g., for decoding a normal television signal on a frame basis, but also to a decoder for decoding an arbitrary shaped video object (e.g., a Video Object Plane disclosed in ISO/IEC JTC1/SC29/WG11/N1902) as a unit without limiting a prediction segment to a fixed-size

block.

Industrial Feasibility

[0196] As discussed above, the memory areas provided for storing reference pictures according to this embodiment enables the adaptive use of the memories for storing data based upon the characteristic of the moving picture sequence. In addition, the contents of one or more of the memory areas can be updated at an arbitrary timing, so that the content of a time-unvarying picture, such as a background picture, is controlled to be updated on a longer-term basis, and the contents of a locally changing picture segment is controlled to be updated on a regular or sequential basis. This achieves an efficient prediction by reflecting the past record of the moving picture sequence.

[0197] Further, the transform parameter value ranges are assigned to the memory areas for making the respective memory areas effective, and the memory areas are switched to be used for prediction among them based upon the value of the transform parameter of a prediction picture segment, thereby achieving an efficient prediction in response to the magnitude of a local/global motion of the moving picture sequence. At the same time, the motion parameters to be encoded on a prediction picture segment basis can be encoded efficiently within the effective motion parameter value ranges of the reference memory areas.

[0198] Further, to the respective memory areas, the transform methods becoming effective in the respective memories are assigned, and the memories are switched to be used for prediction among them in response to the type of the transform parameter of a prediction picture segment, thereby achieving an efficient prediction in response to the complexity of a local/global motion of the moving picture sequence. At the same time, the transform method can be selected adaptively in response to the characteristic of a prediction picture segment, thereby achieving an efficient encoding of the motion parameter.

Claims

1. A moving picture prediction system for predicting a moving picture to be implemented in at least one of an encoder and a decoder, the moving picture prediction system comprising:

   a plurality of memories for storing picture data for reference to be used for prediction, the plurality of memories corresponding to different transform methods, respectively; and
   a prediction picture generation section for receiving a parameter representing a motion of a picture segment to be predicted, and for generating a predicted picture using the picture data stored in one of the plurality of memories used for the picture segment to be predicted based upon the parameter and one of the transform methods corresponding to the one of the plurality of memories.

2. The moving picture prediction system of claim 1, wherein the encoder generates a prediction memory indication information signal indicating the one of the plurality of memories used for generating the predicted picture and transmits the prediction memory indication information signal and the parameter to a decoding station so as to generate the predicted picture using the picture data stored in the one of the plurality of memories based upon the one of the transform methods corresponding to the one of the plurality of memories in the decoding station.

3. The moving picture prediction system of claim 1, wherein the decoder receives the parameter and a prediction memory indication information sinal indicating the one of the plurality of memories used for generating the predicted picture from an encoding station,

   wherein the prediction picture generation section generates the predicted picture using the picture data stored in the one of the plurality of memories based upon the parameter and the one of the transform methods corresponding to the one of the plurality of memories.

4. A moving picture prediction system for predicting a moving picture to be implemented in at least one of an encoding and a decoding, the moving picture prediction system comprising;

   a plurality of memories for storing picture data for reference to be used for prediction, the plurality of memories being assigned to different parameter effective value ranges, respectively; and
   a prediction picture generation section for receiving a parameter representing a motion of a picture segment to be predicted, for selecting one of the plurality of memories assigned to one of the parameter effective value ranges including a value of the parameter, and for generating a predicted picture using the picture data stored in a selected memory.

5. A moving picture prediction system for predicting a moving picture to be implemented in at least one of an encoding and a decoding, the moving picture prediction system comprising:

   a plurality of memories for storing picture data for reference to be used for prediction; and
   a prediction picture generation section includ-

ing

a motion compensator for receiving a parameter representing a motion of a picture segment to be predicted, and for generating a predicted picture by using the picture data stored in the plurality of memories based upon the parameter, and
a memory update unit for updating the picture data stored in at least one of the plurality of memories at an arbitrary timing.

6. The moving picture prediction system of claim 1, 4, or 5, wherein the moving picture is predicted in a moving picture sequence having first and second video objects,

wherein the plurality of memories includes separate first and second pluralities of memories corresponding to the first and second video objects, respectively, and the prediction picture generation section includes separate first and second generators, respectively, corresponding to the first and second video objects,

wherein the first generator uses the picture data stored in at least one of the first and second pluralities of memories to generate the predicted picture when predicting the first object, and generates information indicating a use of the second plurality of memories for predicting the first object, the information being added to the predicted picture.

7. The moving picture prediction system of claim 1, 4 or 5, wherein the prediction picture generation section generates the predicted picture through a change of either one of a number and a size of the plurality of memories in response to a change in the moving picture at each time instance.

8. The moving picture prediction system of claim 1, 4 or 5, wherein the prediction picture generation section generates the predicted picture in a limited use of memories for prediction in response to a change in the moving picture at each time instance.

9. The moving picture prediction system of claim 1, wherein the prediction picture generation section generates the predicted picture by calculating a plurality of the predicted pictures generated by using the respective picture data stored in the plurality of memories.

10. The moving picture prediction system of claim 1 further comprising:

a significance detector for detecting a feature parameter representing a significance of the picture segment to be predicted,

wherein the prediction picture generation section generates the predicted picture by selecting at least one of choices of at least one of a plurality of prediction methods, the plurality of memories, and a plurality of memory update methods.

11. The moving picture prediction system of claim 1 or 5 further comprising:

a significance detector for detecting a parameter representing at least one of an amount of bits available for coding the picture segment to be predicted, an amount of change of the picture segment at each time instance, and a significance of the picture segment;

wherein the prediction picture generation section generates the predicted picture by selecting at least one of choices of at least one of a plurality of prediction methods, the plurality of memories, a plurality of memory update methods.

12. The moving picture prediction system of claim 1 or 5, wherein the moving picture is predicted on a video object basis,

wherein the moving picture prediction system further comprising:

a significance detector for detecting a parameter representing at least one of an amount of bits available for coding a video object to be predicted, an amount of change in the video object at each time instance, and a significance of the video object;

wherein the prediction picture generation section generates the predicted picture by selecting at least one of choices of at least one of a plurality of prediction methods, the plurality of memories, and a plurality of memory update methods.

13. The moving picture prediction system of claim 1 or 5 further comprising:

a prediction information encoder for encoding prediction relating information of the moving picture;

wherein the prediction picture generation section counts a number of times of a memory used for prediction and determines a rank of the plurality of memories based upon a counted number of the times,

wherein the prediction information encoder allocates a code length to the prediction relating information to be encoded based upon the rank of a memory used for prediction.

14. The moving picture prediction system of claim 1, wherein the plurality of memories includes at least a frame memory for storing the picture data on a frame basis and a sprite memory for storing a sprite picture.

15. The moving picture prediction system of claim 14, wherein the sprite memory includes at least one of a dynamic sprite memory involving a regular updating, and a static sprite memory not involving the regular updating.

16. The moving picture prediction system of claim 14, wherein the one of the transform methods corresponding to the one of the plurality of memories is at least one of a parallel translation, an affine transformation, and a perspective transformation in an interchangeable manner.

17. A method for predicting a moving picture to be implemented in at least one of an encoding or a decoding, the method comprising the steps of:

storing picture data for reference to be used for prediction in a plurality of memories;
corresponding different transform methods with the plurality of memories, respectively;
receiving a parameter representing a motion of a picture segment to be predicted; and
generating a predicted picture using the picture data stored in one of the plurality of memories used for predicting the picture segment based upon the parameter and one of the transform methods being corresponding to the one of the plurality of memories.

18. The method for predicting a moving picture of claim 17, further comprising the steps of:

generating a prediction memory indication information signal indicating the one of the plurality of memories used for the picture segment to be predicted; and
transmitting the prediction memory indication information signal and the parameter to a decoding station.

19. The method for predicting a moving picture of claim 17 to be implemented in the decoding, the method further comprising the step of:

receiving a prediction memory indication information signal indicating the one of the plurality of memories used for generating the predicted picture and the parameter representing a motion of the picture segment to be predicted from an encoding station.

20. A method for predicting a moving picture to be implemented in at least one of an encoding and a decoding, the method comprising the steps of:

storing picture data for reference to be used for prediction in a plurality of memories;
assigning separate parameter effective value ranges to the plurality of memories, respectively;
receiving a parameter representing a motion of a picture segment to be predicted;
selecting one of the plurality of memories assigned to one of the parameter effective value ranges including a value of the parameter; and
generating a predicted picture using the picture data stored in a selected memory.

21. A method for predicting a moving picture to be implemented in at least one of an encoding and a decoding, the method comprising the steps of:

storing picture data for reference to be used for prediction in a plurality of memories;
receiving a parameter representing a motion of a picture segment to be predicted;
generating a predicted picture using the picture data stored in the plurality of memories based upon the parameter; and
updating the picture data stored in at least one of the plurality of memories at an arbitrary timing.

Fig.1

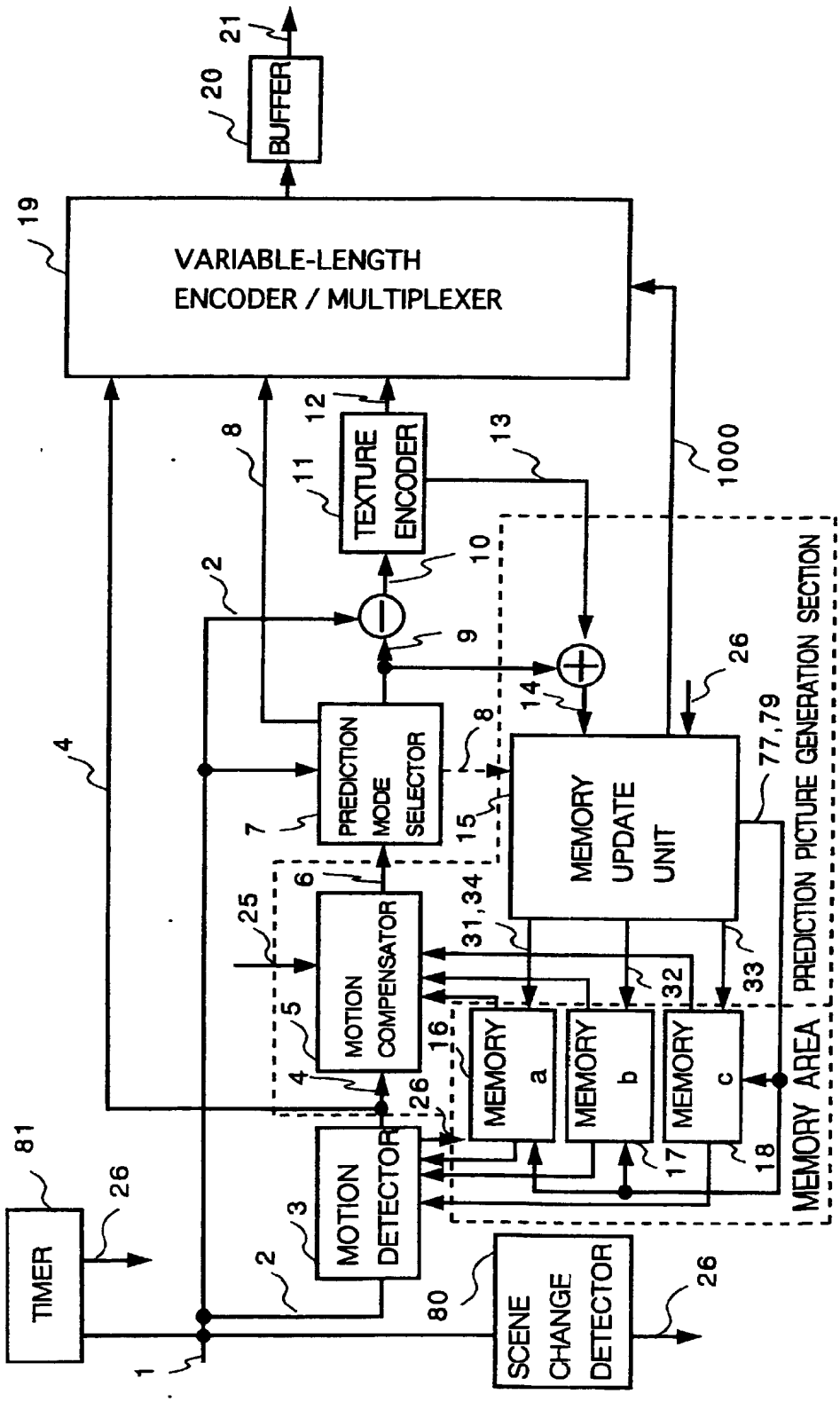

BUFFER
VARIABLE-LENGTH ENCODER / MULTIPLEXER
TEXTURE ENCODER
PREDICTION MODE SELECTOR
MOTION COMPENSATOR
MOTION DETECTOR
MEMORY a
MEMORY b
MEMORY c
MEMORY UPDATE UNIT
SCENE CHANGE DETECTOR
TIMER
MEMORY AREA
PREDICTION PICTURE GENERATION SECTION

Fig.2

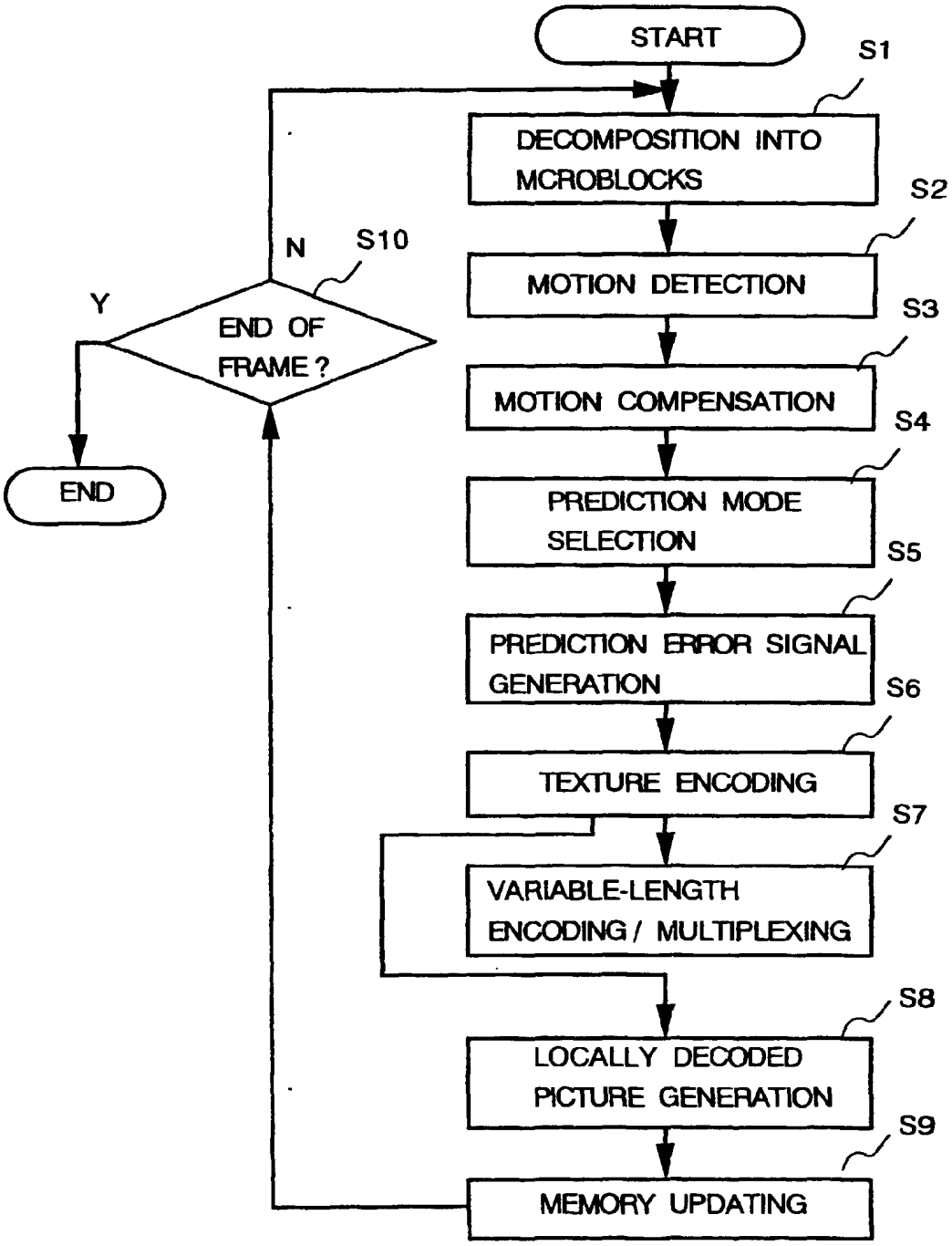

START
S1
DECOMPOSITION INTO MCROBLOCKS
S2
MOTION DETECTION
S3
MOTION COMPENSATION
S4
PREDICTION MODE SELECTION
S5
PREDICTION ERROR SIGNAL GENERATION
S6
TEXTURE ENCODING
S7
VARIABLE-LENGTH ENCODING / MULTIPLEXING
S8
LOCALLY DECODED PICTURE GENERATION
S9
MEMORY UPDATING
S10
END OF FRAME ?
N
Y
END

Fig.3

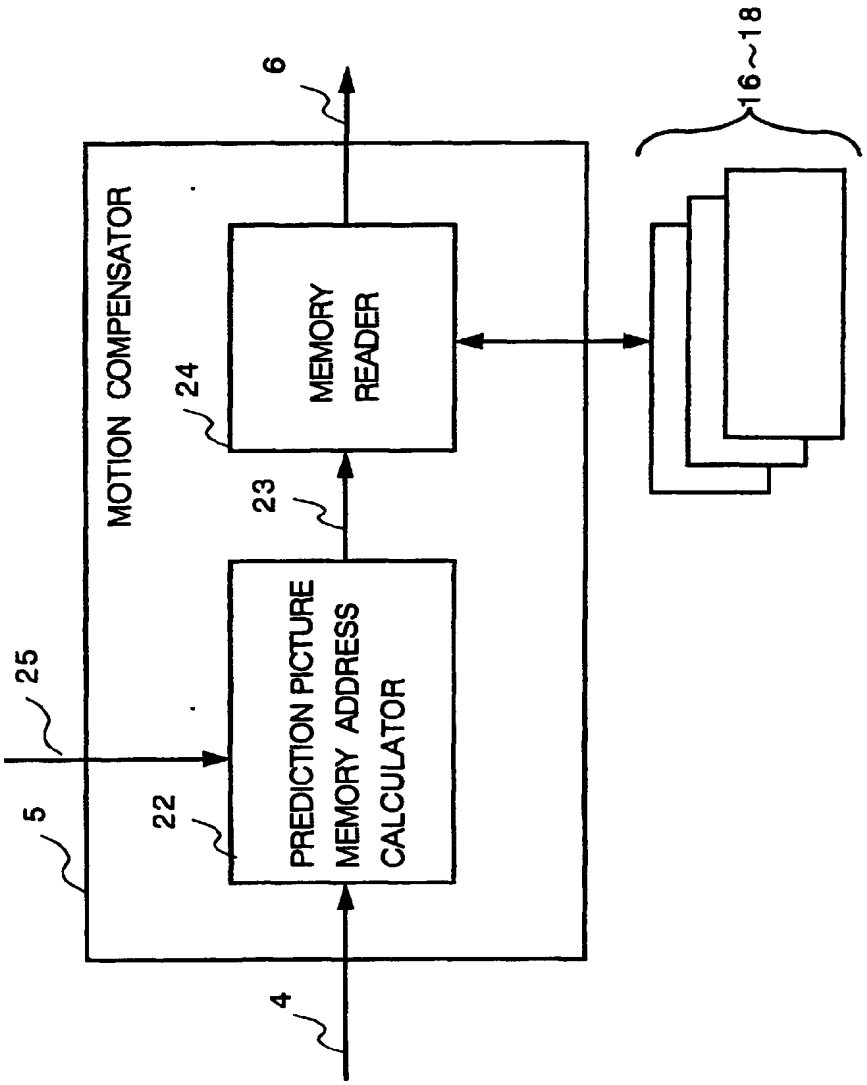

16~18
6
MOTION COMPENSATOR
24
MEMORY READER
23
PREDICTION PICTURE MEMORY ADDRESS CALCULATOR
22
25
5
4

Fig.4

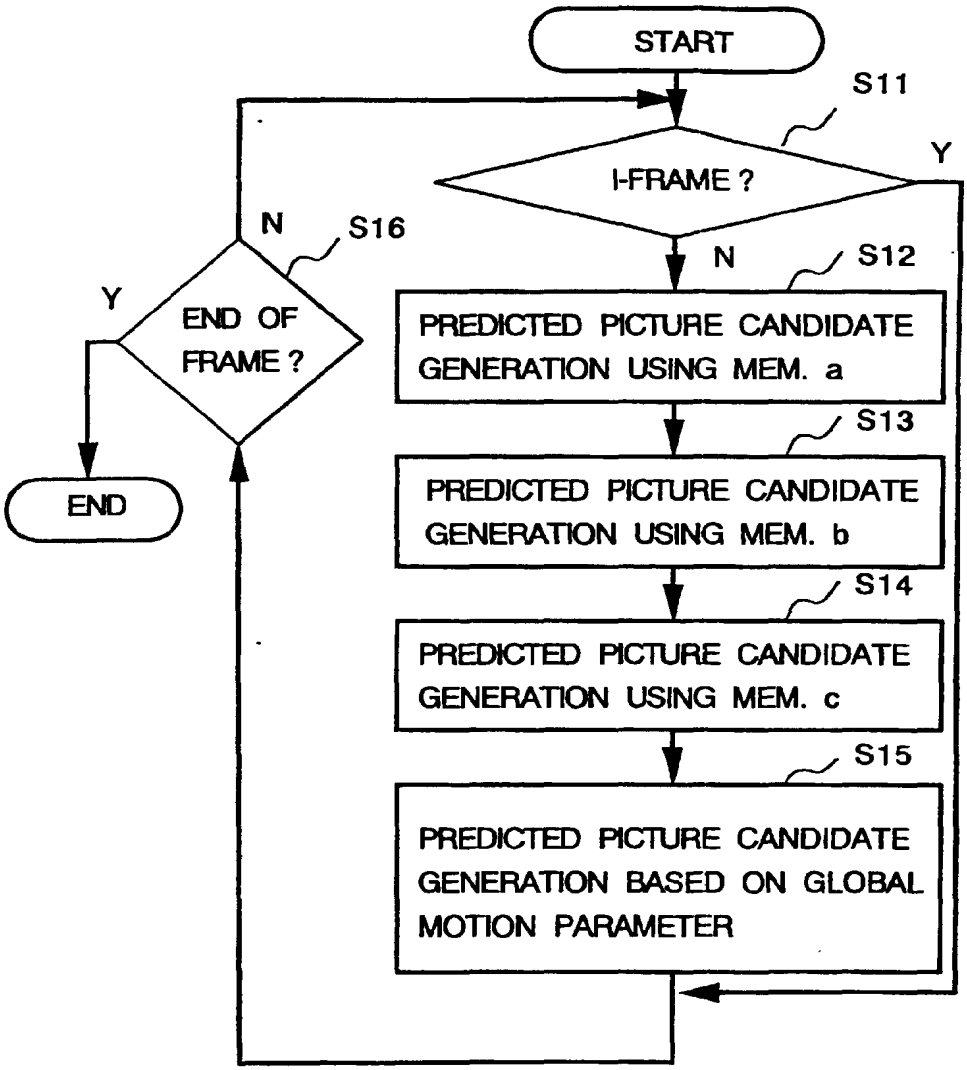

START
S11
I-FRAME ?
Y
N
S12
PREDICTED PICTURE CANDIDATE GENERATION USING MEM. a
S13
PREDICTED PICTURE CANDIDATE GENERATION USING MEM. b
S14
PREDICTED PICTURE CANDIDATE GENERATION USING MEM. c
S15
PREDICTED PICTURE CANDIDATE GENERATION BASED ON GLOBAL MOTION PARAMETER
N
S16
END OF FRAME ?
Y
END

Fig.5

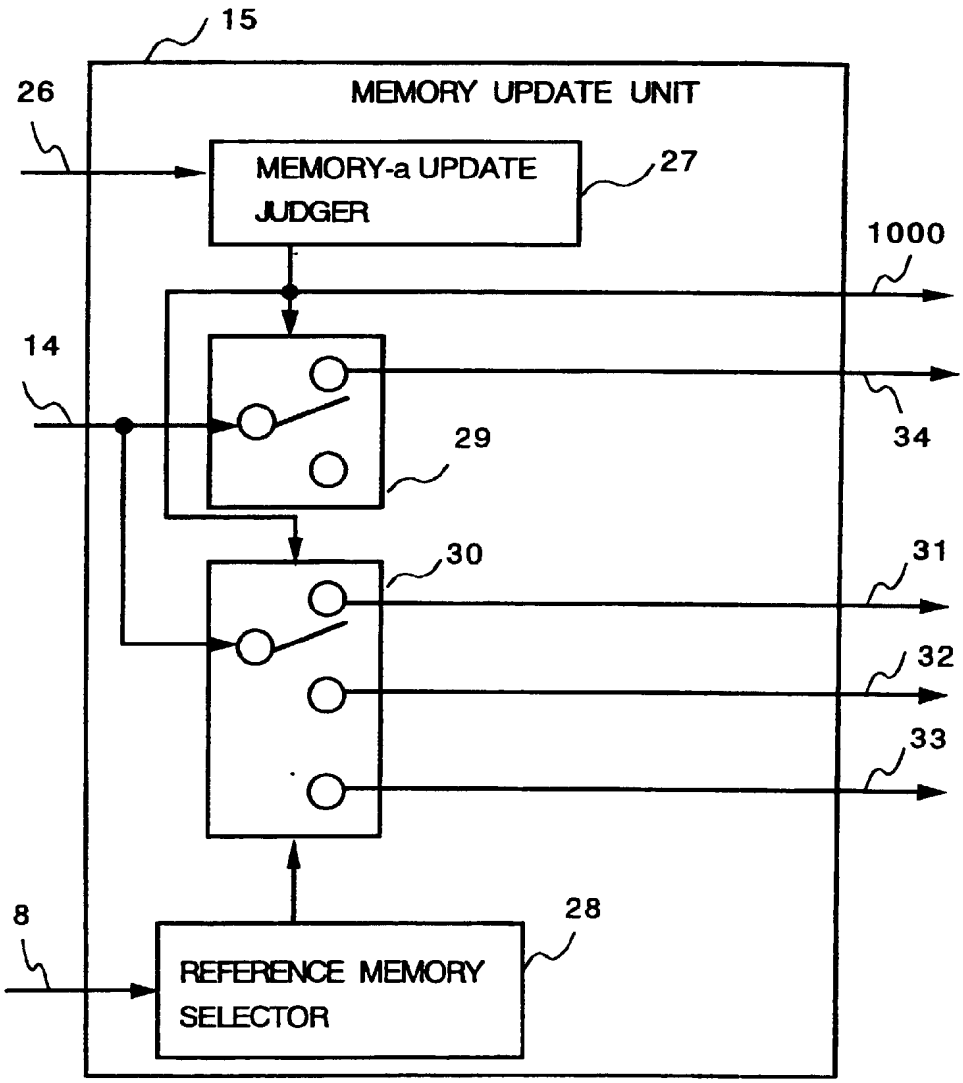

15
MEMORY UPDATE UNIT
26
MEMORY-a UPDATE JUDGER
27
1000
14
34
29
30
31
32
33
8
28
REFERENCE MEMORY SELECTOR

Fig.6

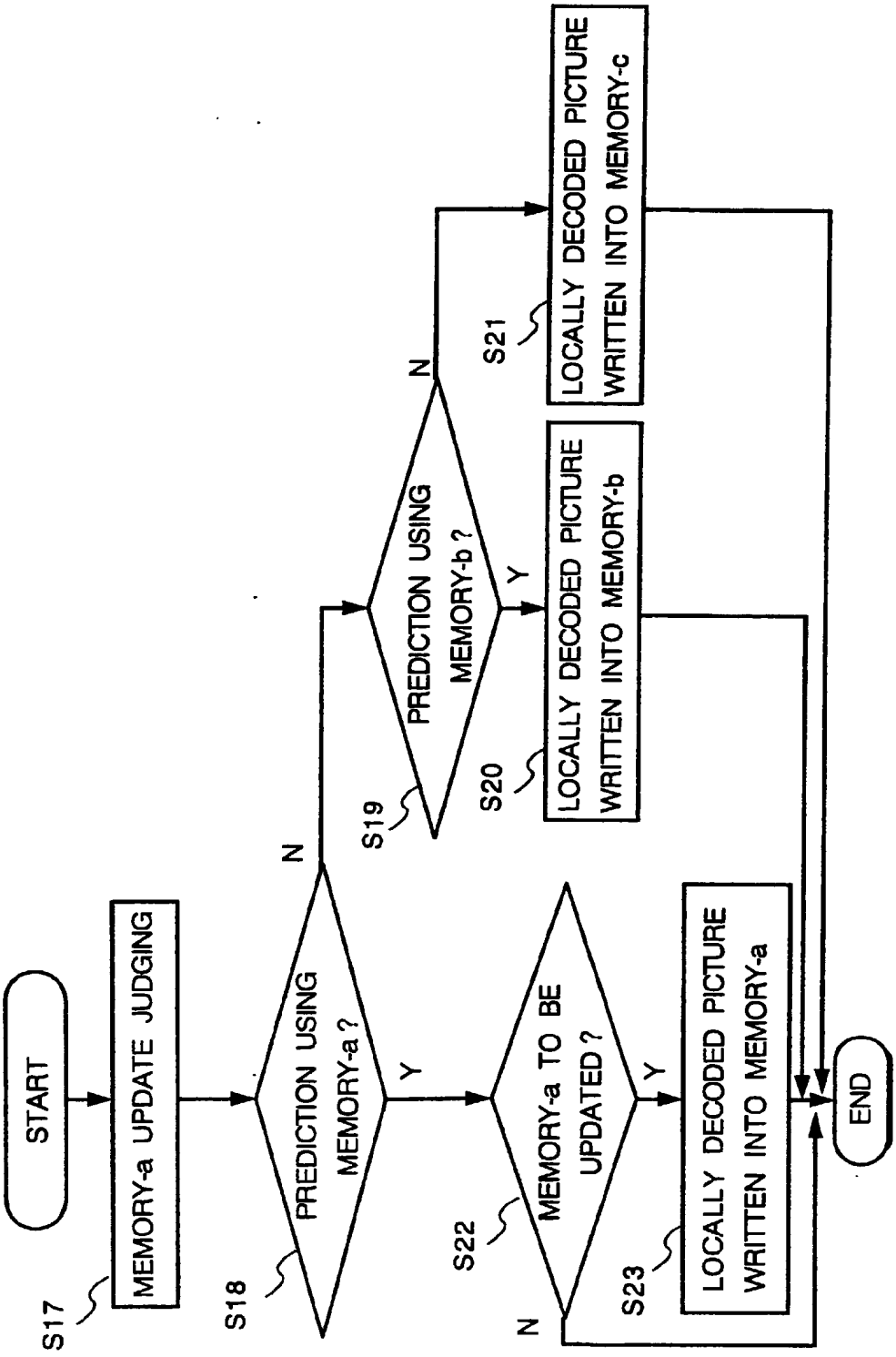

START
MEMORY-a UPDATE JUDGING
S17
PREDICTION USING MEMORY-a ?
S18
Y
N
PREDICTION USING MEMORY-b ?
S19
Y
N
LOCALLY DECODED PICTURE WRITTEN INTO MEMORY-b
S20
LOCALLY DECODED PICTURE WRITTEN INTO MEMORY-c
S21
MEMORY-a TO BE UPDATED ?
S22
Y
N
LOCALLY DECODED PICTURE WRITTEN INTO MEMORY-a
S23
END

Fig.7

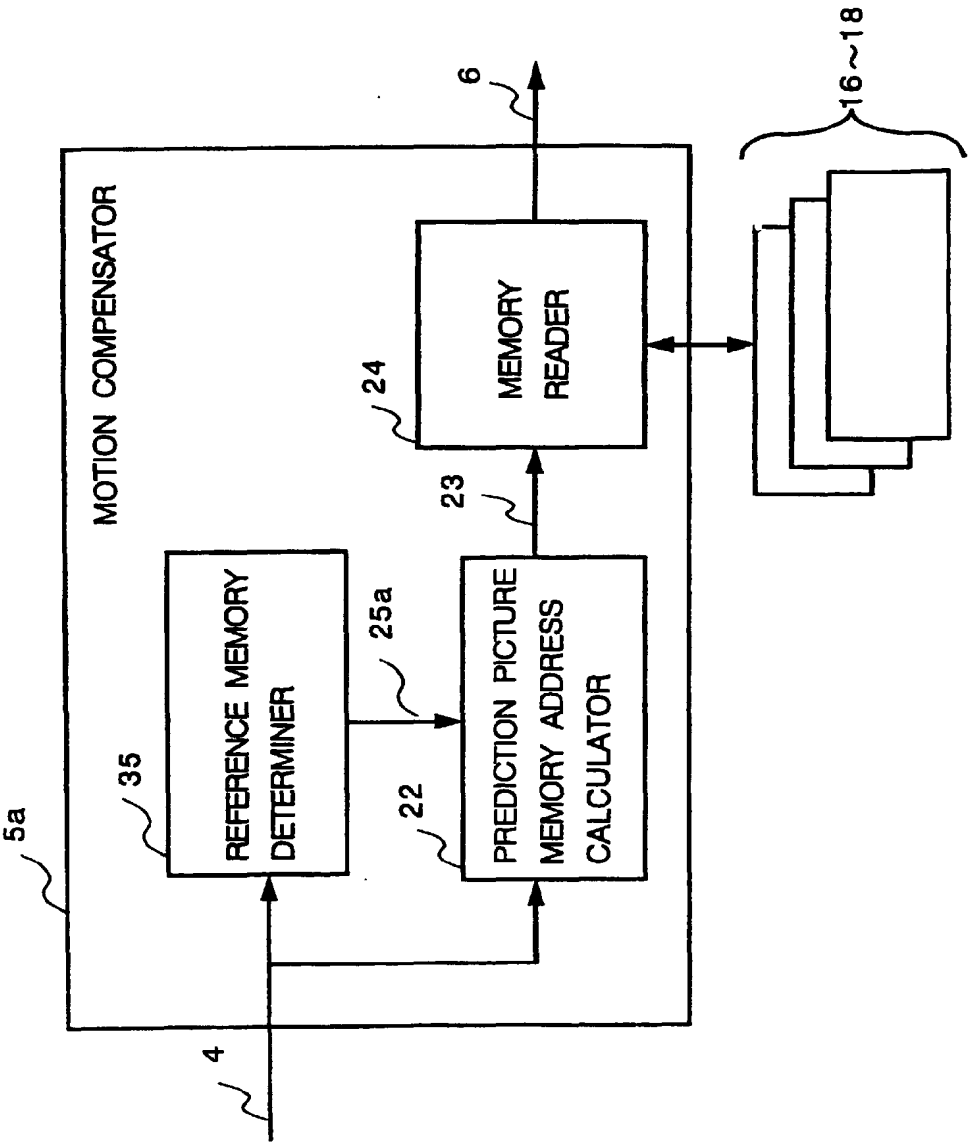

MOTION COMPENSATOR
5a
35
REFERENCE MEMORY DETERMINER
25a
22
PREDICTION PICTURE MEMORY ADDRESS CALCULATOR
23
24
MEMORY READER
6
16~18
4

Fig.8

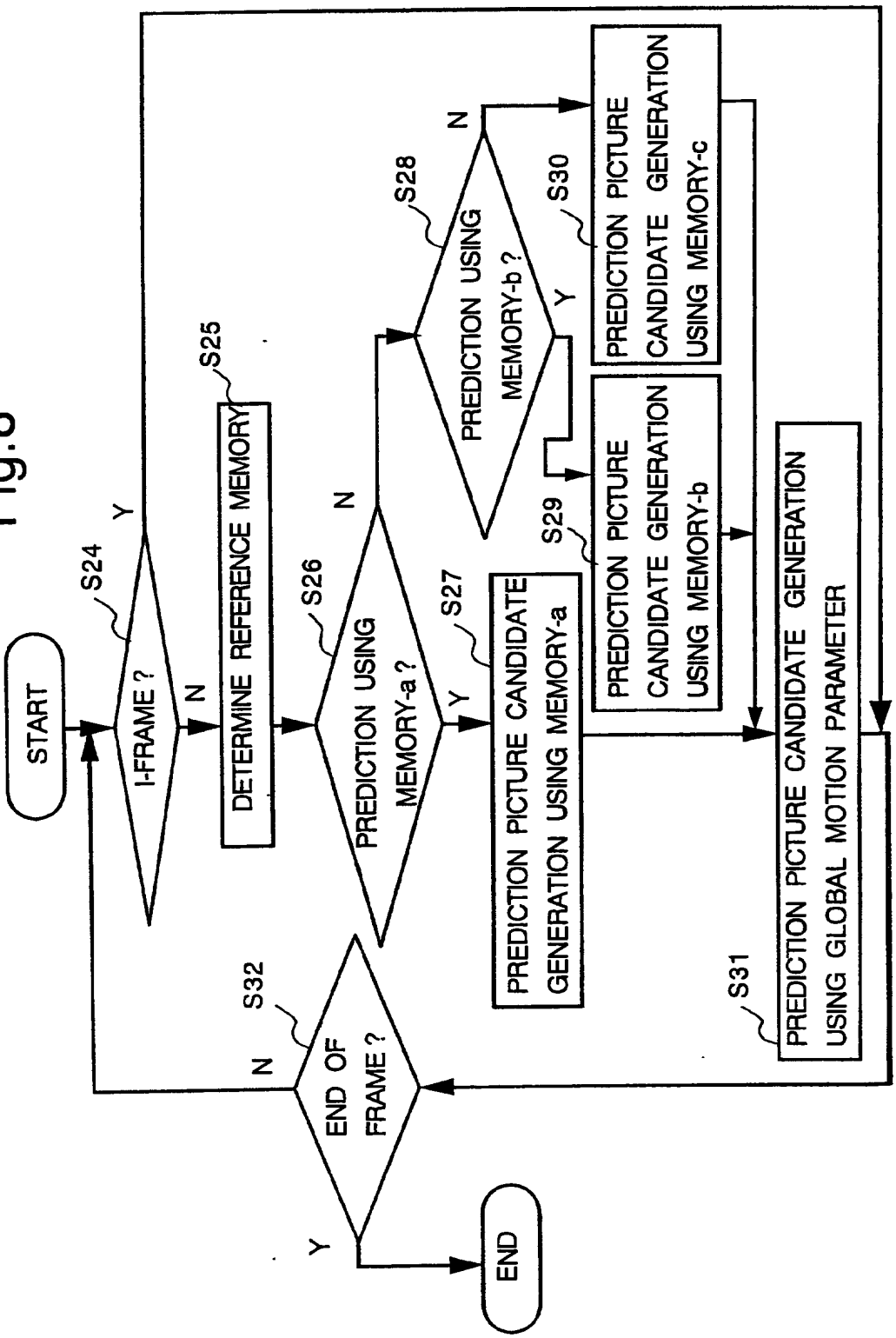

START
I-FRAME ?
Y
N
S24
DETERMINE REFERENCE MEMORY
S25
PREDICTION USING MEMORY-a ?
N
Y
S26
PREDICTION USING MEMORY-b ?
N
Y
S28
PREDICTION PICTURE CANDIDATE GENERATION USING MEMORY-a
S27
PREDICTION PICTURE CANDIDATE GENERATION USING MEMORY-b
S29
PREDICTION PICTURE CANDIDATE GENERATION USING MEMORY-c
S30
PREDICTION PICTURE CANDIDATE GENERATION USING GLOBAL MOTION PARAMETER
S31
END OF FRAME ?
N
Y
S32
END

Fig.9

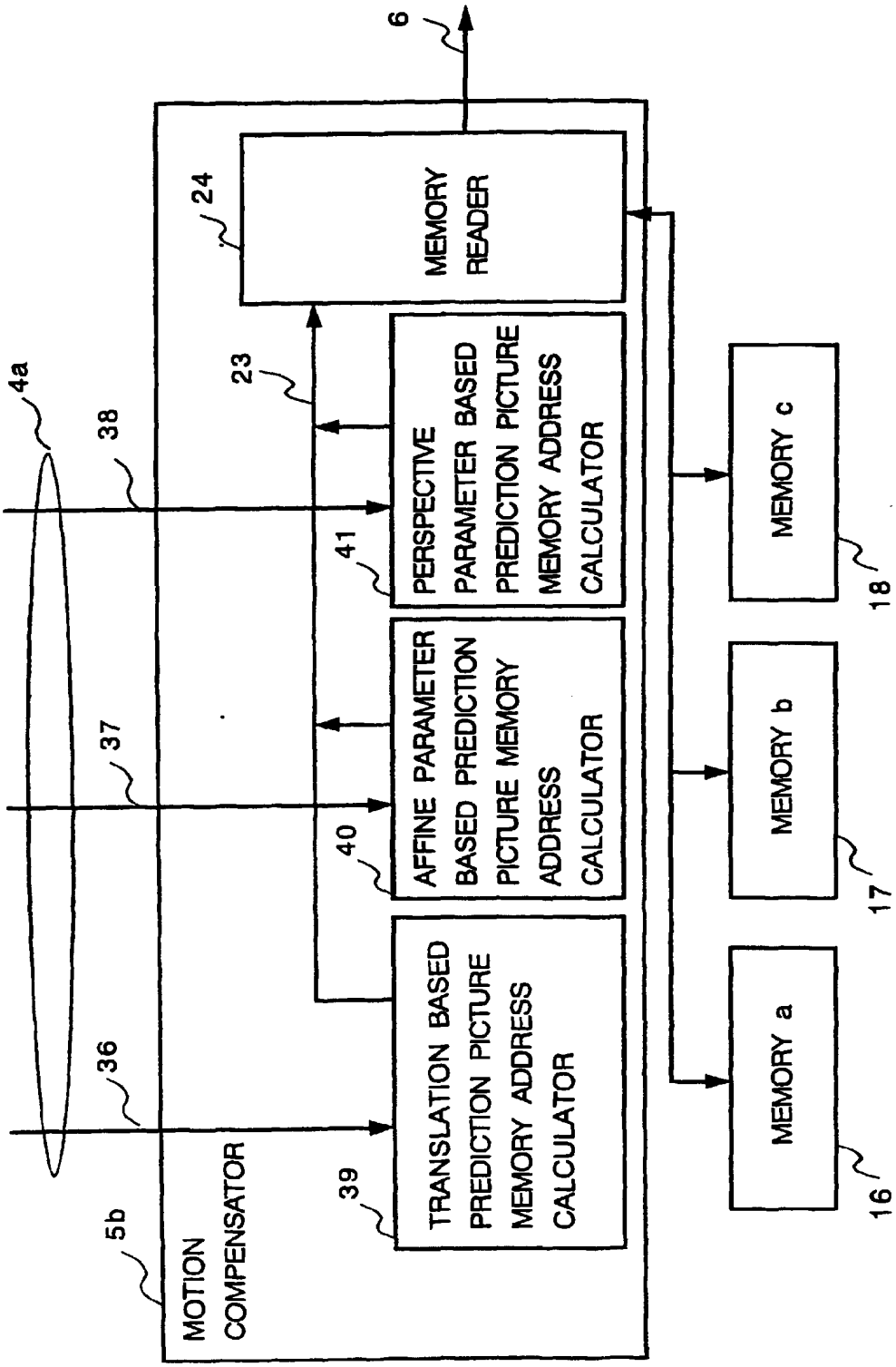

6
24
MEMORY READER
23
PERSPECTIVE PARAMETER BASED PREDICTION PICTURE MEMORY ADDRESS CALCULATOR
41
AFFINE PARAMETER BASED PREDICTION PICTURE MEMORY ADDRESS CALCULATOR
40
TRANSLATION BASED PREDICTION PICTURE MEMORY ADDRESS CALCULATOR
39
MOTION COMPENSATOR
5b
4a
38
37
36
MEMORY a
16
MEMORY b
17
MEMORY c
18

Fig.10

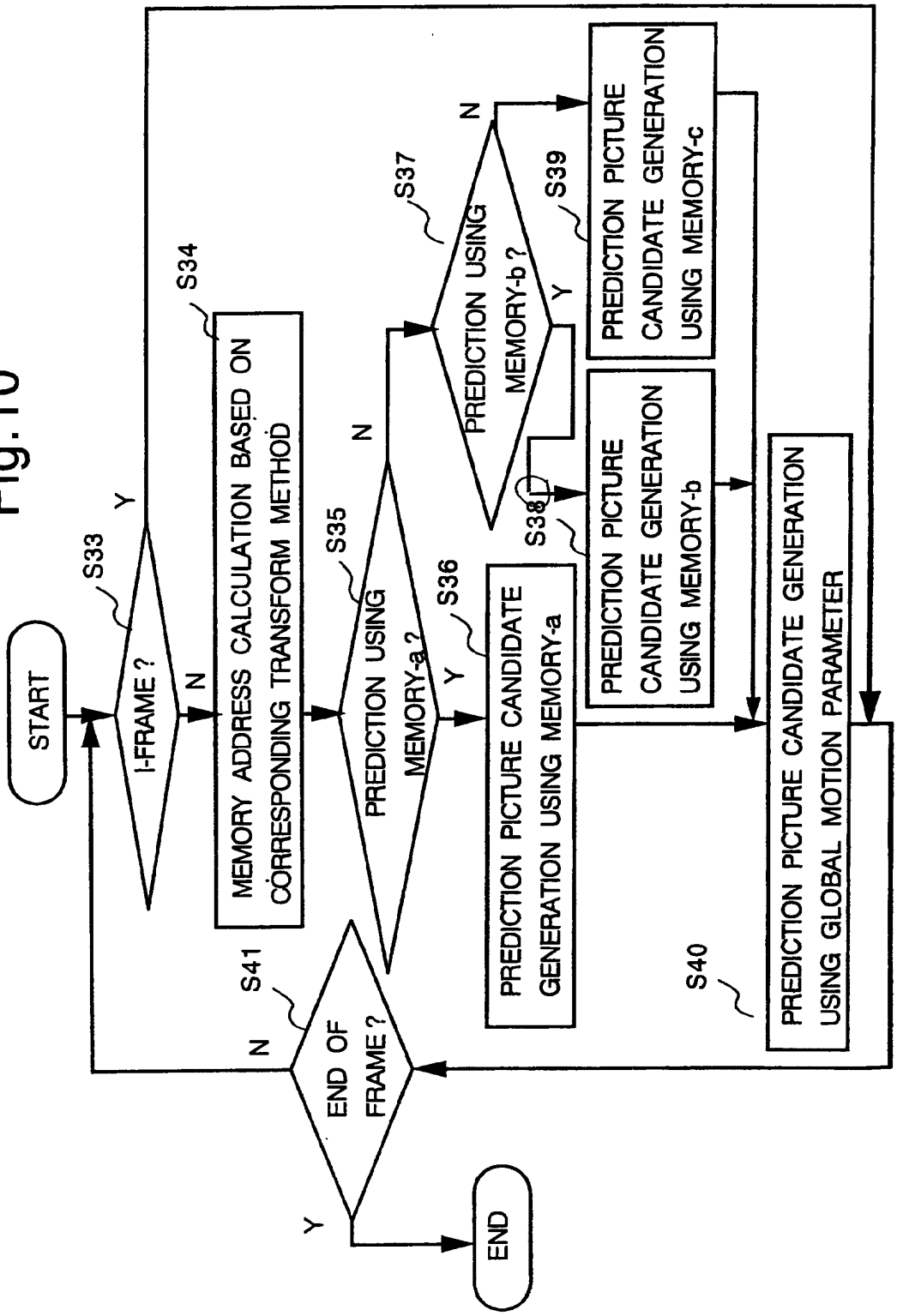

START
I-FRAME ?
Y
N
S33
MEMORY ADDRESS CALCULATION BASED ON CORRESPONDING TRANSFORM METHOD
S34
PREDICTION USING MEMORY-a ?
N
Y
S35
PREDICTION PICTURE CANDIDATE GENERATION USING MEMORY-a
S36
PREDICTION USING MEMORY-b ?
N
Y
S37
PREDICTION PICTURE CANDIDATE GENERATION USING MEMORY-b
S38
PREDICTION PICTURE CANDIDATE GENERATION USING MEMORY-c
S39
PREDICTION PICTURE CANDIDATE GENERATION USING GLOBAL MOTION PARAMETER
S40
END OF FRAME ?
N
Y
S41
END

Fig.11

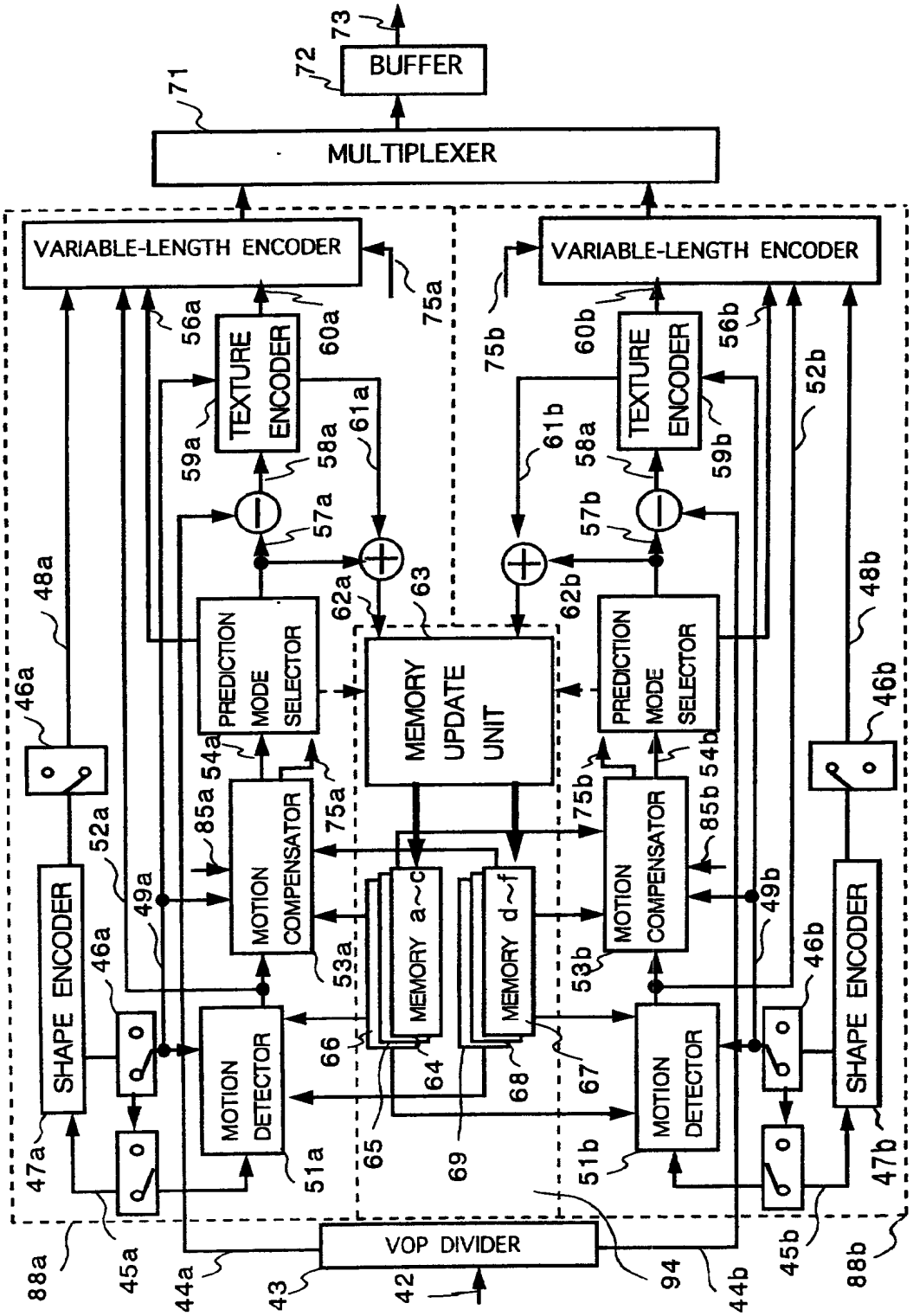

BUFFER
MULTIPLEXER
VARIABLE-LENGTH ENCODER
VARIABLE-LENGTH ENCODER
TEXTURE ENCODER
TEXTURE ENCODER
PREDICTION MODE SELECTOR
PREDICTION MODE SELECTOR
MEMORY UPDATE UNIT
MOTION COMPENSATOR
MOTION COMPENSATOR
MEMORY a~c
MEMORY d~f
MOTION DETECTOR
MOTION DETECTOR
SHAPE ENCODER
SHAPE ENCODER
VOP DIVIDER

Fig.12

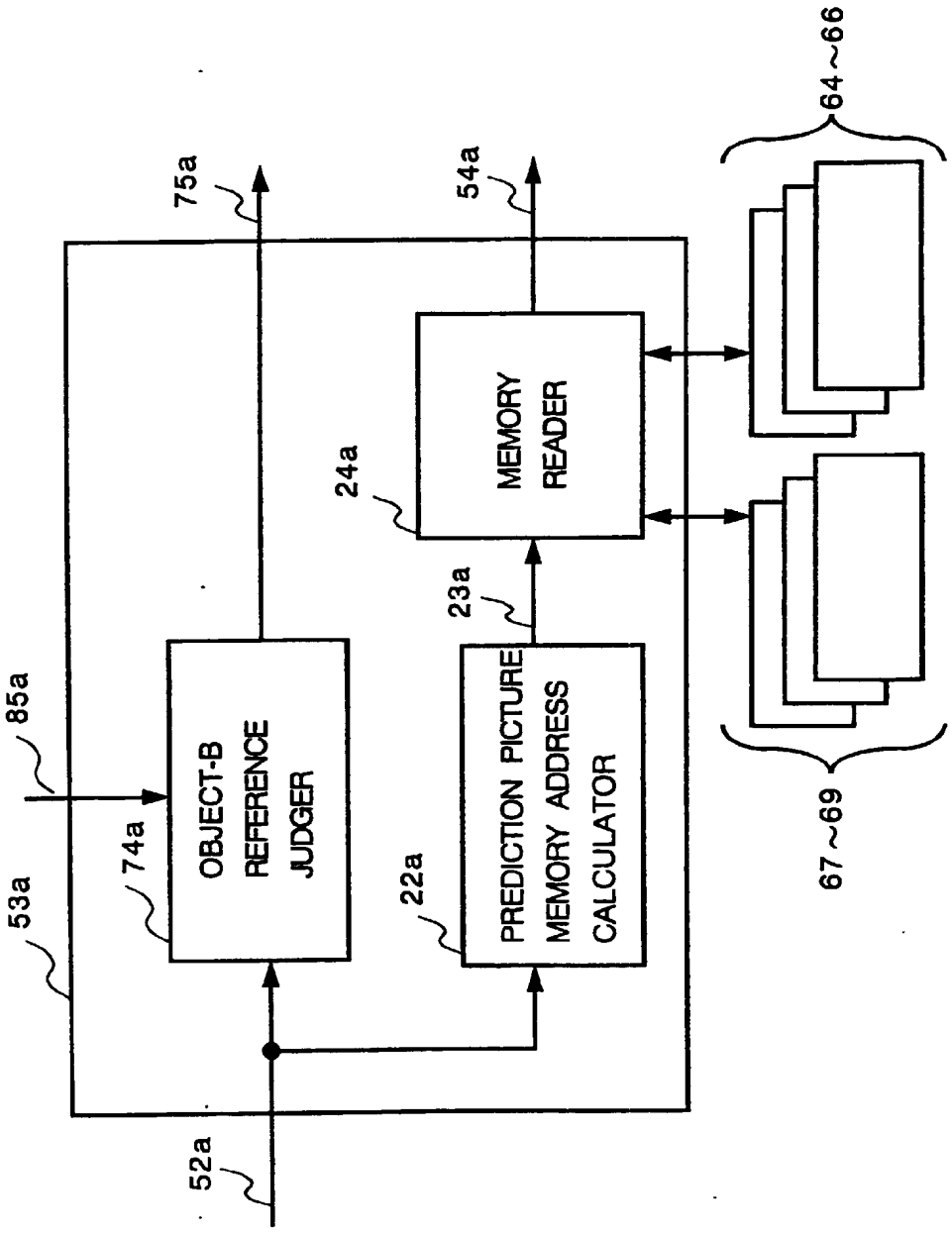

64~66
75a
54a
MEMORY READER
24a
23a
85a
OBJECT-B REFERENCE JUDGER
74a
PREDICTION PICTURE MEMORY ADDRESS CALCULATOR
22a
53a
67~69
52a

Fig.13

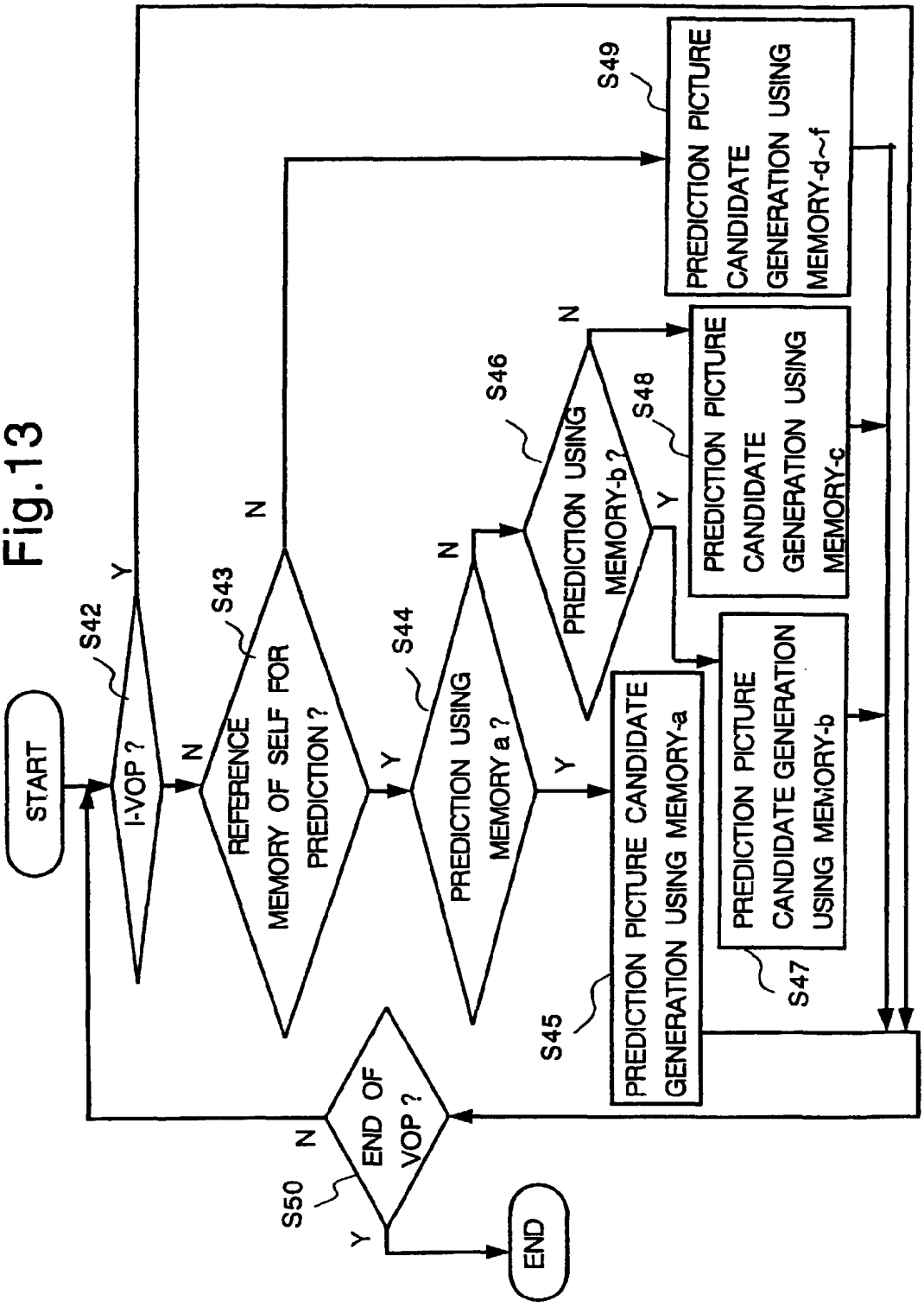

START
I-VOP ?
Y
N
S42
REFERENCE MEMORY OF SELF FOR PREDICTION ?
Y
N
S43
PREDICTION USING MEMORY a ?
Y
N
S44
PREDICTION USING MEMORY-b ?
Y
N
S46
S48
PREDICTION PICTURE CANDIDATE GENERATION USING MEMORY-a
S45
PREDICTION PICTURE CANDIDATE GENERATION USING MEMORY-b
S47
PREDICTION PICTURE CANDIDATE GENERATION USING MEMORY-c
PREDICTION PICTURE CANDIDATE GENERATION USING MEMORY-d~f
S49
END OF VOP ?
N
Y
S50
END

Fig.14

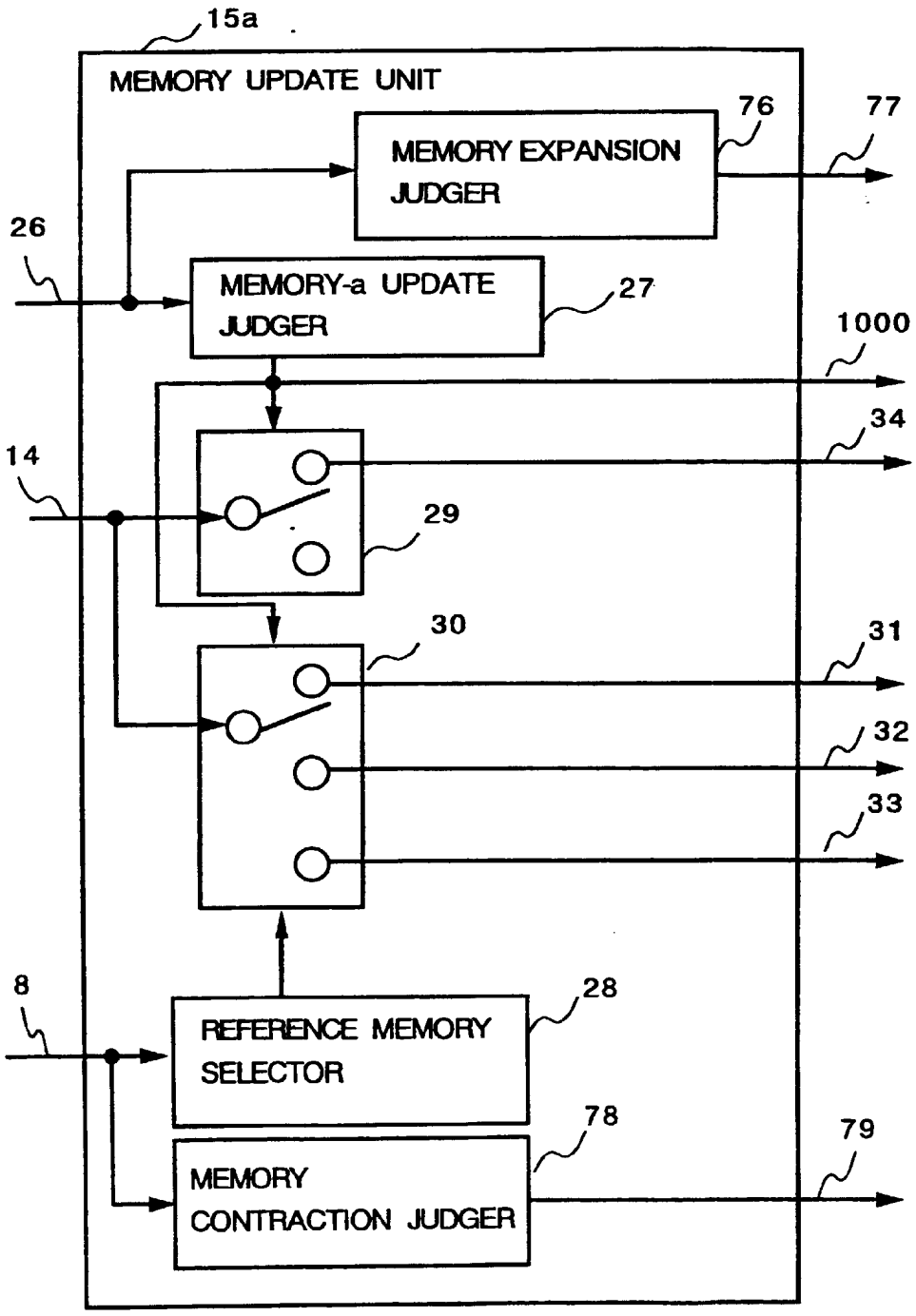

15a
MEMORY UPDATE UNIT
76
77
MEMORY EXPANSION JUDGER
26
MEMORY-a UPDATE JUDGER
27
1000
34
14
29
30
31
32
33
8
REFERENCE MEMORY SELECTOR
28
78
79
MEMORY CONTRACTION JUDGER

Fig.15

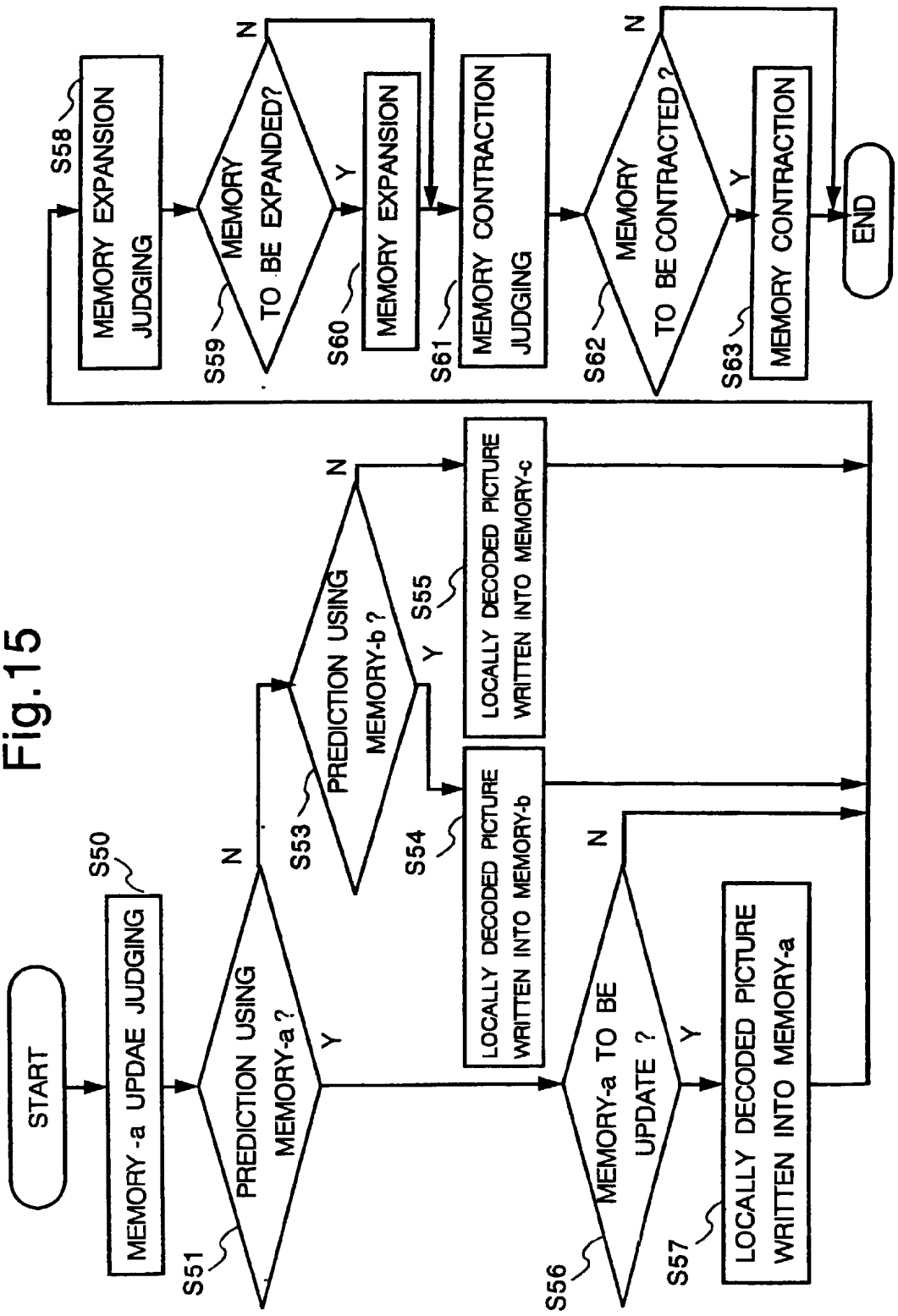

START
MEMORY-a UPDAE JUDGING
S50
PREDICTION USING MEMORY-a ?
S51
N
Y
PREDICTION USING MEMORY-b ?
S53
N
Y
LOCALLY DECODED PICTURE WRITTEN INTO MEMORY-b
S54
LOCALLY DECODED PICTURE WRITTEN INTO MEMORY-c
S55
MEMORY-a TO BE UPDATE ?
S56
N
Y
LOCALLY DECODED PICTURE WRITTEN INTO MEMORY-a
S57
MEMORY EXPANSION JUDGING
S58
MEMORY TO BE EXPANDED?
S59
N
Y
MEMORY EXPANSION
S60
MEMORY CONTRACTION JUDGING
S61
MEMORY TO BE CONTRACTED ?
S62
N
Y
MEMORY CONTRACTION
S63
END

Fig.16

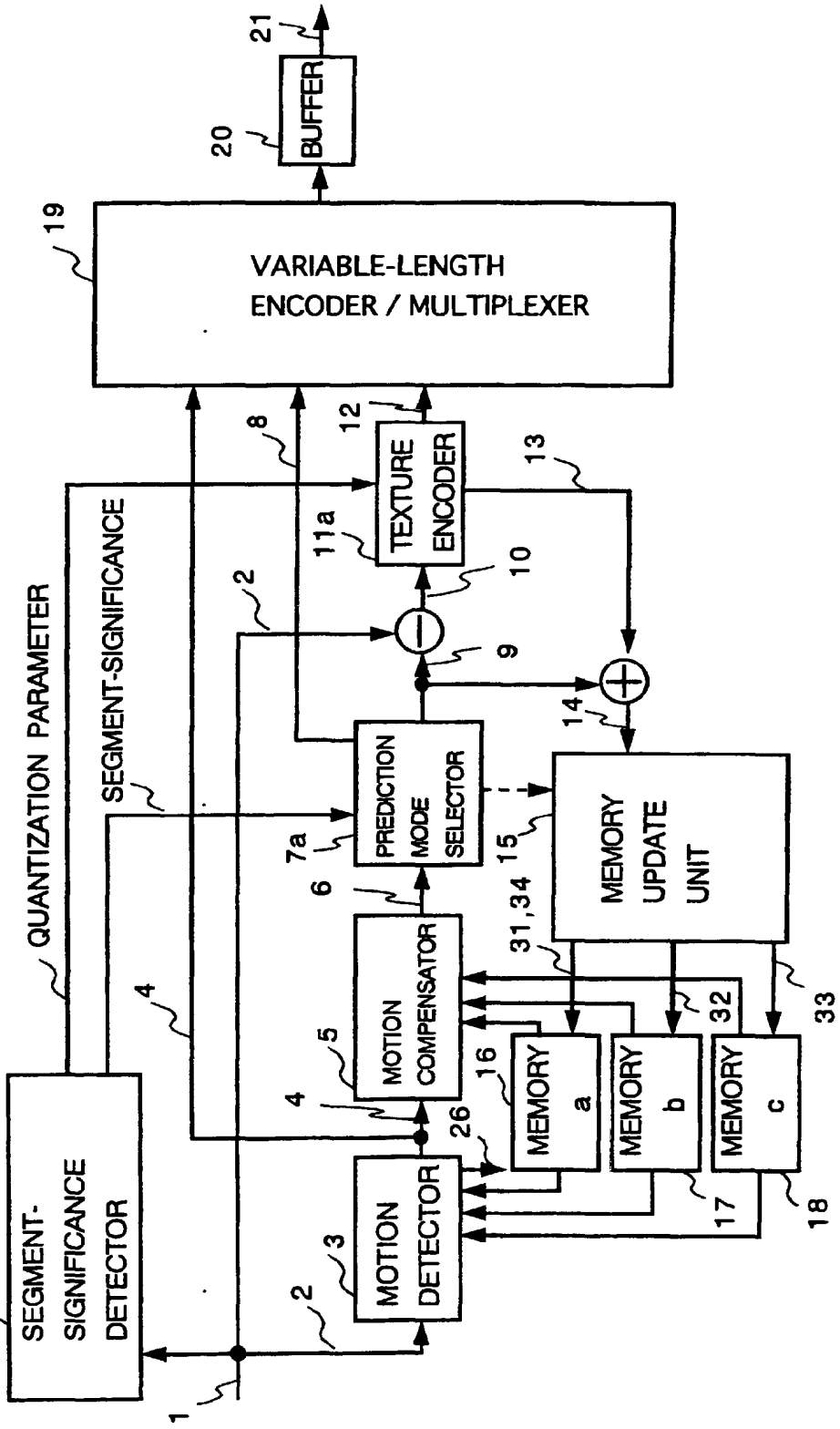

21
BUFFER
20
19
VARIABLE-LENGTH
ENCODER / MULTIPLEXER
8
12
TEXTURE
ENCODER
13
11a
10
2
9
14
QUANTIZATION PARAMETER
SEGMENT-SIGNIFICANCE
PREDICTION
MODE
SELECTOR
7a
6
15
MEMORY
UPDATE
UNIT
31,34
4
MOTION
COMPENSATOR
5
32
33
16
MEMORY
a
MEMORY
b
MEMORY
c
4
26
17
18
95
SEGMENT-
SIGNIFICANCE
DETECTOR
MOTION
DETECTOR
3
2
1

Fig.17

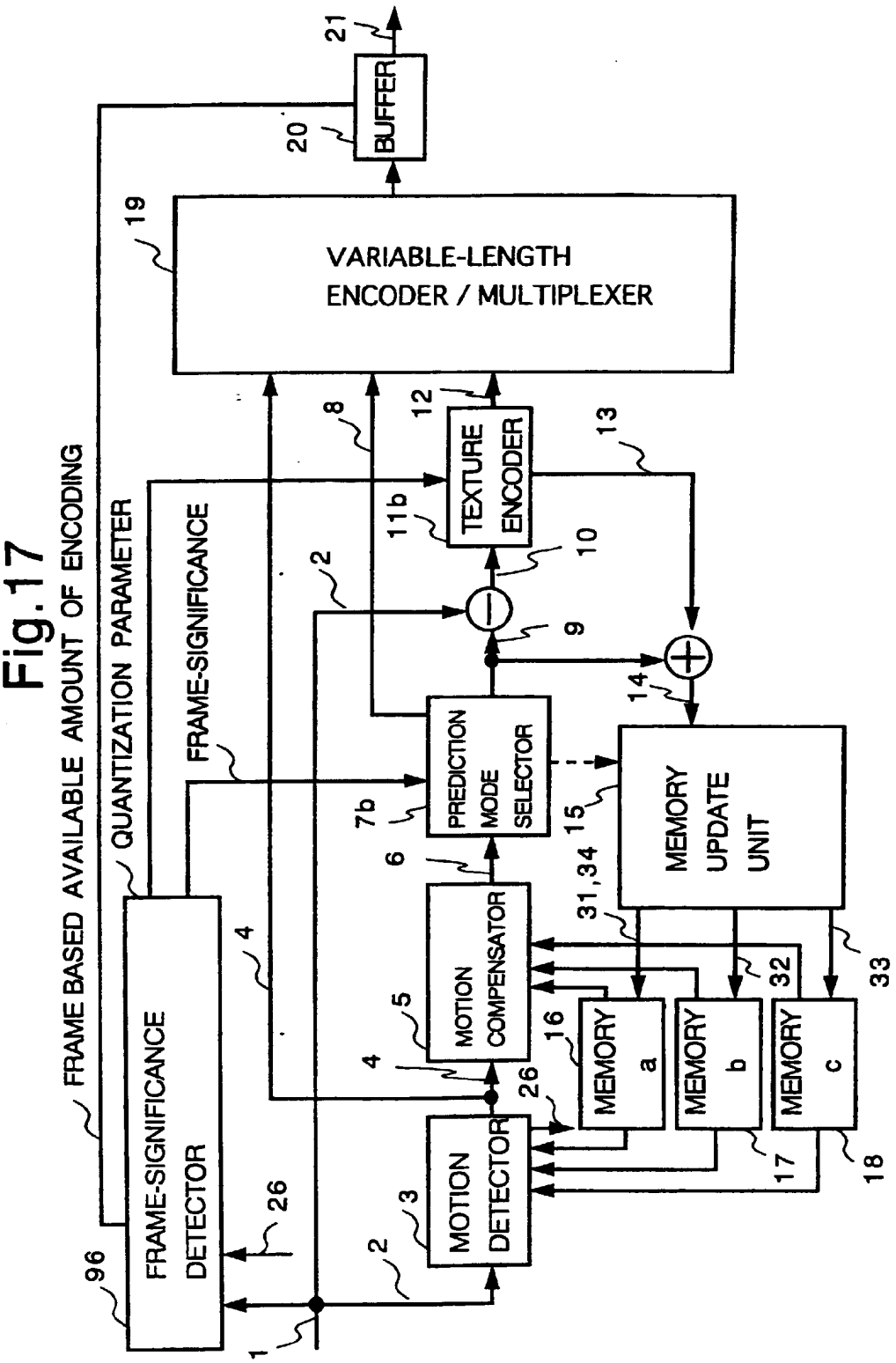

FRAME BASED AVAILABLE AMOUNT OF ENCODING
QUANTIZATION PARAMETER
FRAME-SIGNIFICANCE
FRAME-SIGNIFICANCE DETECTOR
MOTION DETECTOR
MOTION COMPENSATOR
PREDICTION MODE SELECTOR
TEXTURE ENCODER
VARIABLE-LENGTH ENCODER / MULTIPLEXER
BUFFER
MEMORY UPDATE UNIT
MEMORY a
MEMORY b
MEMORY c
96
1
2
3
4
5
6
7b
8
9
10
11b
12
13
14
15
16
17
18
19
20
21
26
31,34
32
33

Fig.18

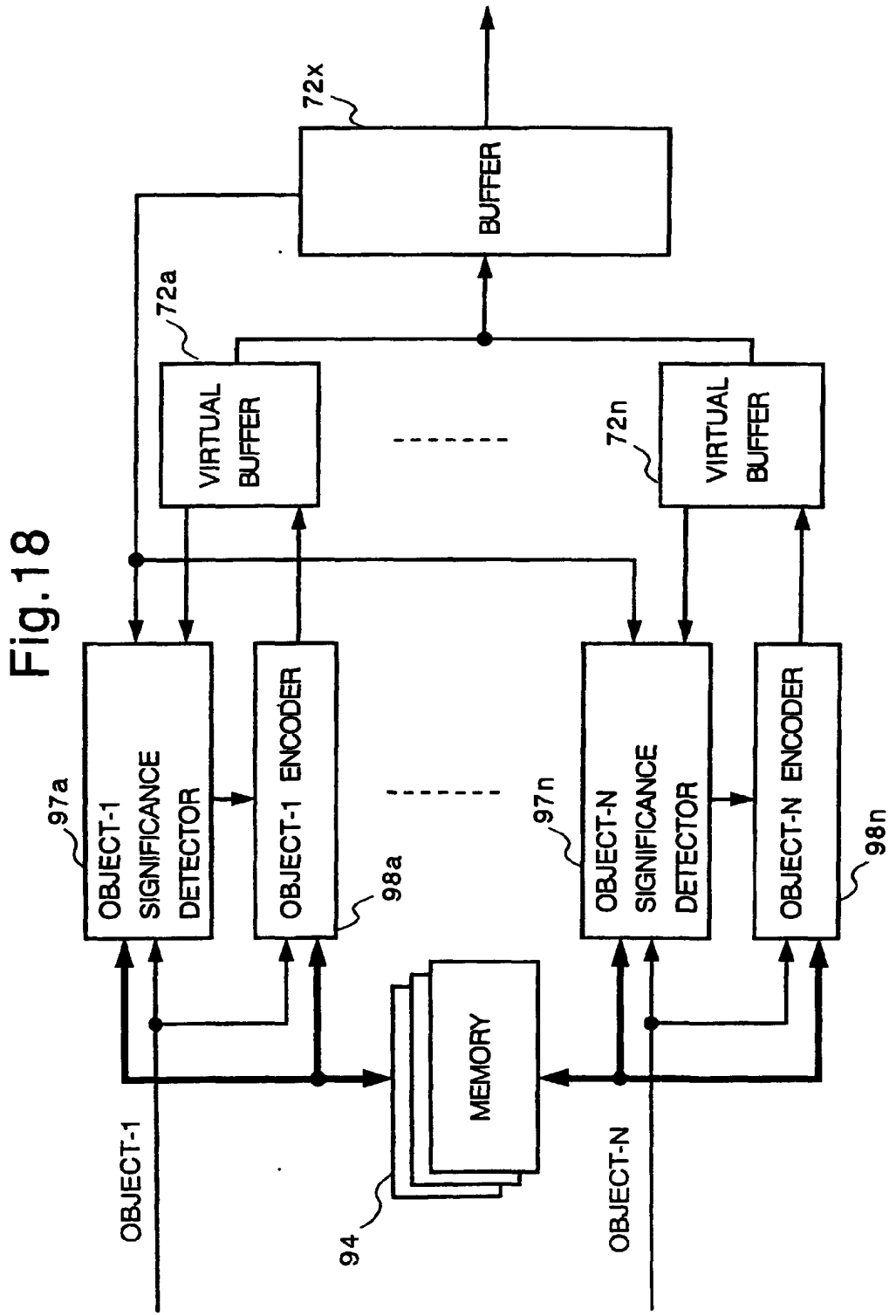

72x
BUFFER
72a
VIRTUAL BUFFER
72n
VIRTUAL BUFFER
97a
OBJECT-1 SIGNIFICANCE DETECTOR
OBJECT-1 ENCODER
98a
97n
OBJECT-N SIGNIFICANCE DETECTOR
OBJECT-N ENCODER
98n
MEMORY
94
OBJECT-1
OBJECT-N

Fig.19

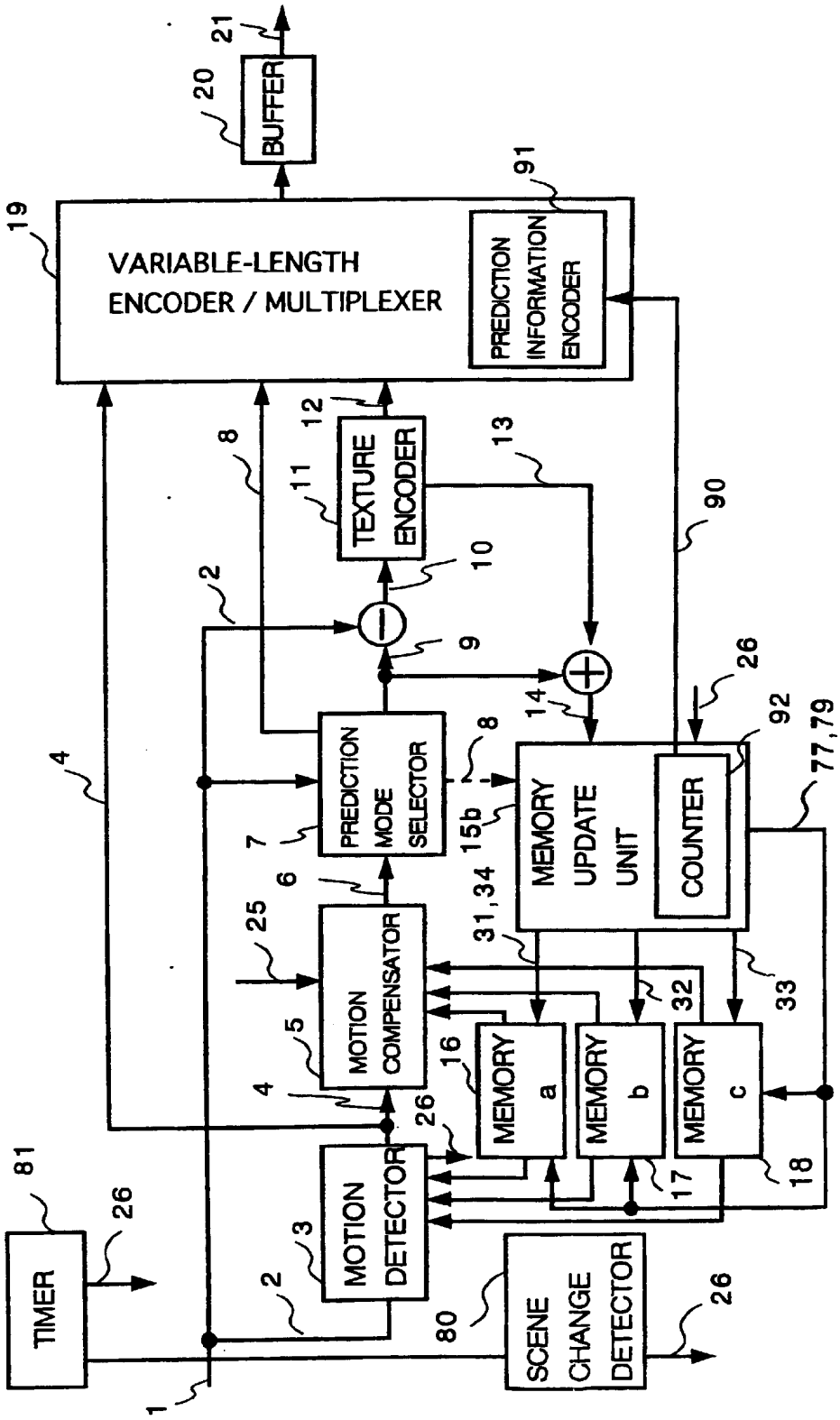

BUFFER
VARIABLE-LENGTH ENCODER / MULTIPLEXER
PREDICTION INFORMATION ENCODER
TEXTURE ENCODER
PREDICTION MODE SELECTOR
MOTION COMPENSATOR
MOTION DETECTOR
MEMORY UPDATE UNIT
COUNTER
MEMORY a
MEMORY b
MEMORY c
TIMER
SCENE CHANGE DETECTOR

Fig. 20

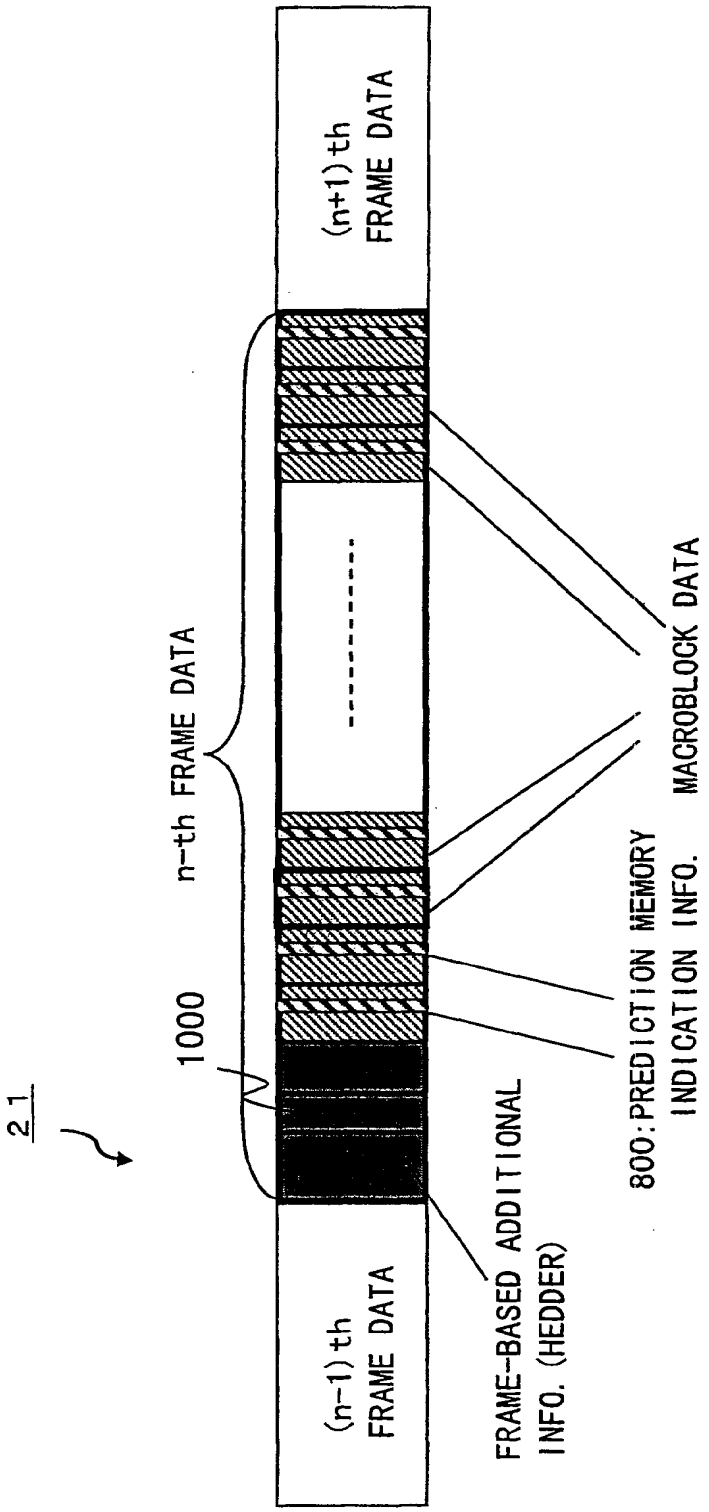

21
1000
n-th FRAME DATA
(n+1)th FRAME DATA
(n-1)th FRAME DATA
FRAME-BASED ADDITIONAL INFO. (HEDDER)
800: PREDICTION MEMORY INDICATION INFO.
MACROBLOCK DATA

Fig. 21

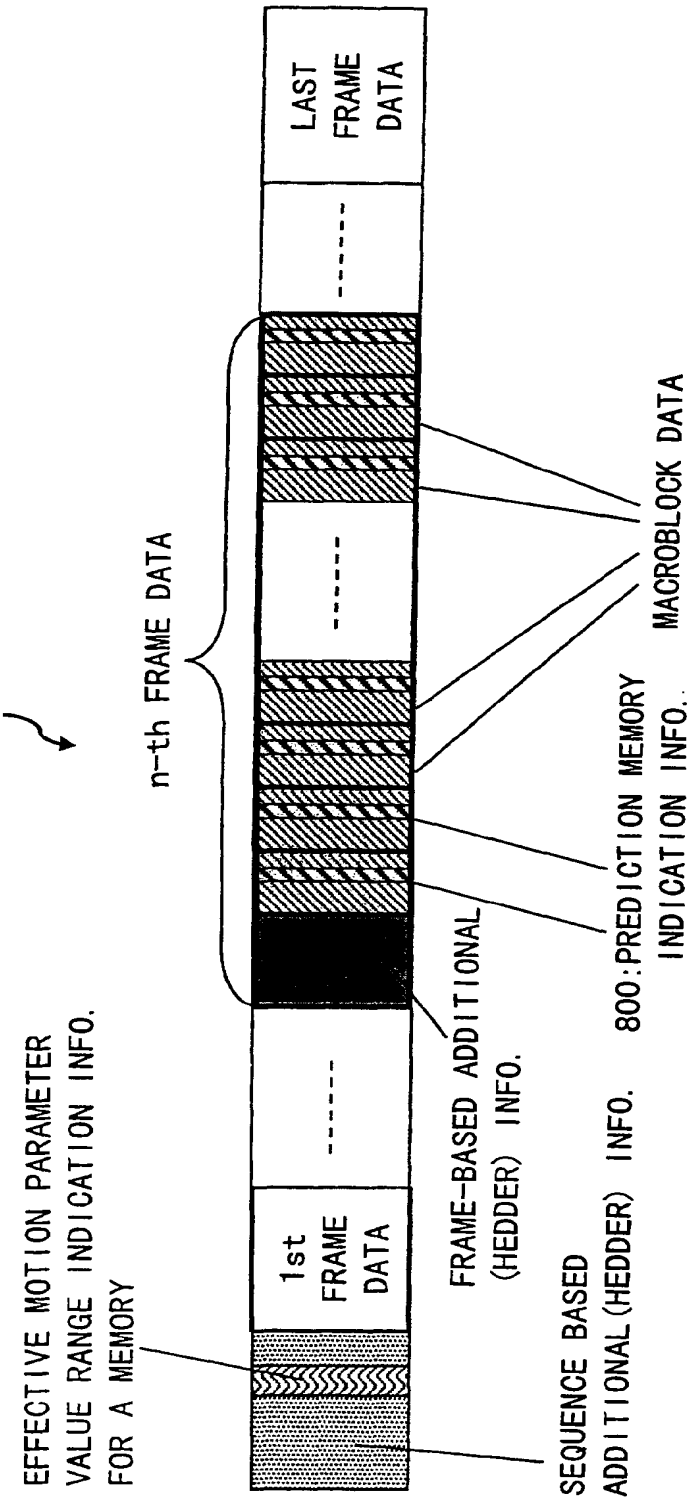

21
EFFECTIVE MOTION PARAMETER VALUE RANGE INDICATION INFO. FOR A MEMORY
SEQUENCE BASED ADDITIONAL(HEDDER) INFO.
1st FRAME DATA
n-th FRAME DATA
LAST FRAME DATA
FRAME-BASED ADDITIONAL (HEDDER) INFO.
800:PREDICTION MEMORY INDICATION INFO.
MACROBLOCK DATA

Fig. 22

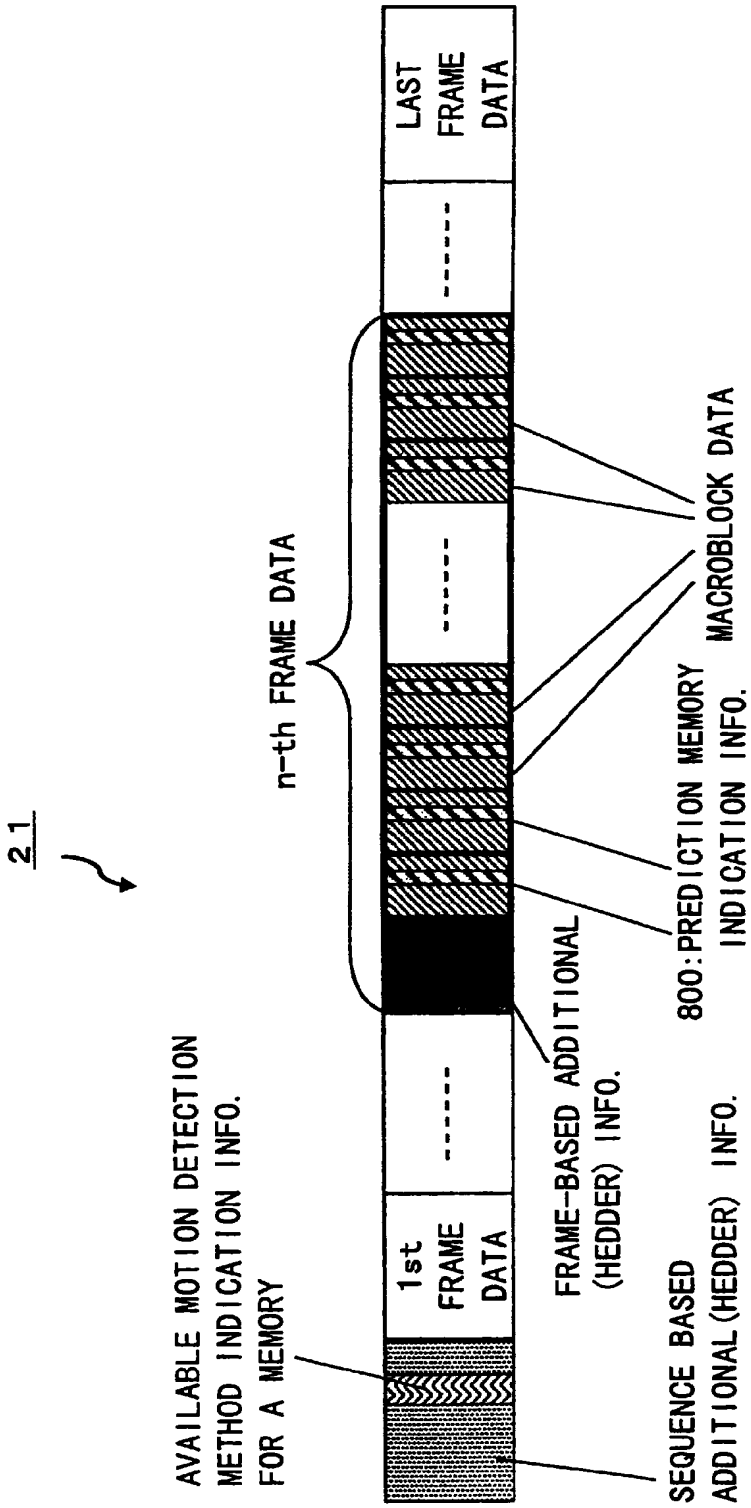

21
n-th FRAME DATA
1st FRAME DATA
LAST FRAME DATA
AVAILABLE MOTION DETECTION METHOD INDICATION INFO. FOR A MEMORY
FRAME-BASED ADDITIONAL (HEDDER) INFO.
SEQUENCE BASED ADDITIONAL (HEDDER) INFO.
800: PREDICTION MEMORY INDICATION INFO.
MACROBLOCK DATA

Fig. 23

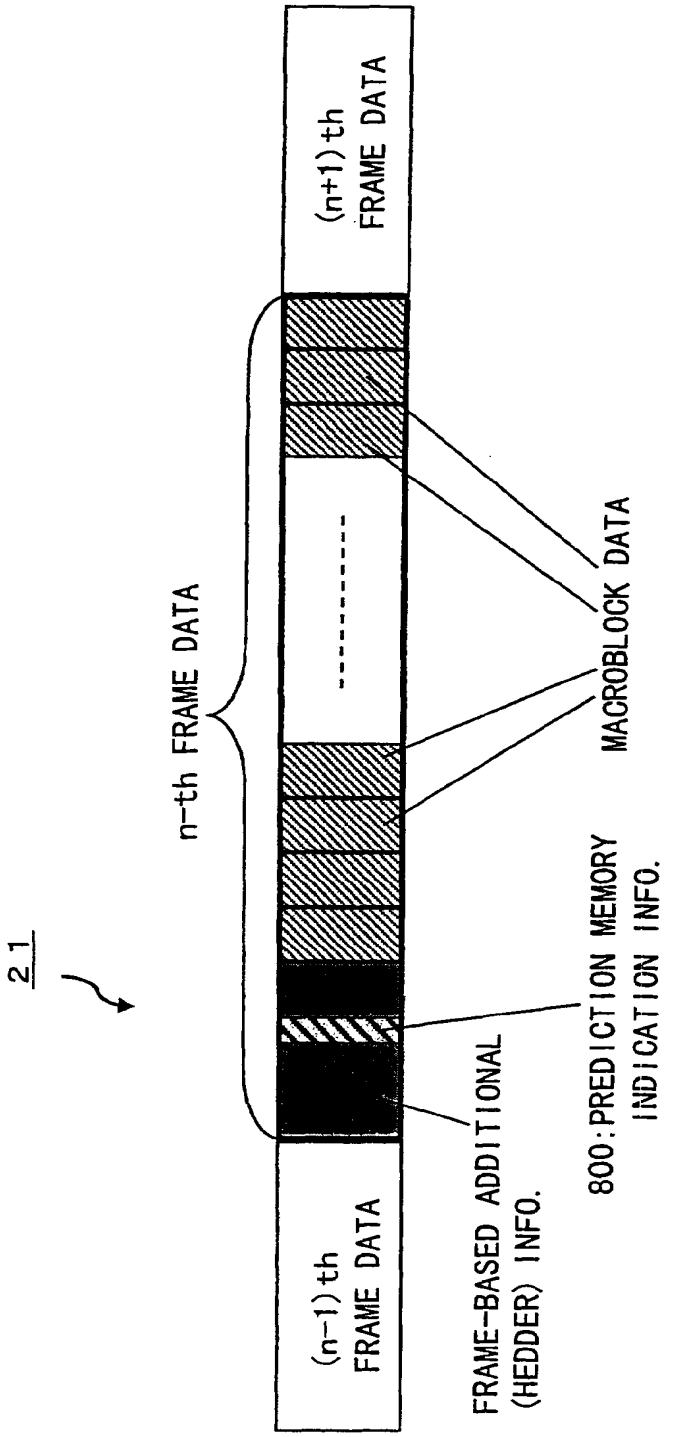

21
n-th FRAME DATA
(n+1)th FRAME DATA
(n-1)th FRAME DATA
MACROBLOCK DATA
800:PREDICTION MEMORY INDICATION INFO.
FRAME-BASED ADDITIONAL (HEDDER) INFO.

Fig. 24

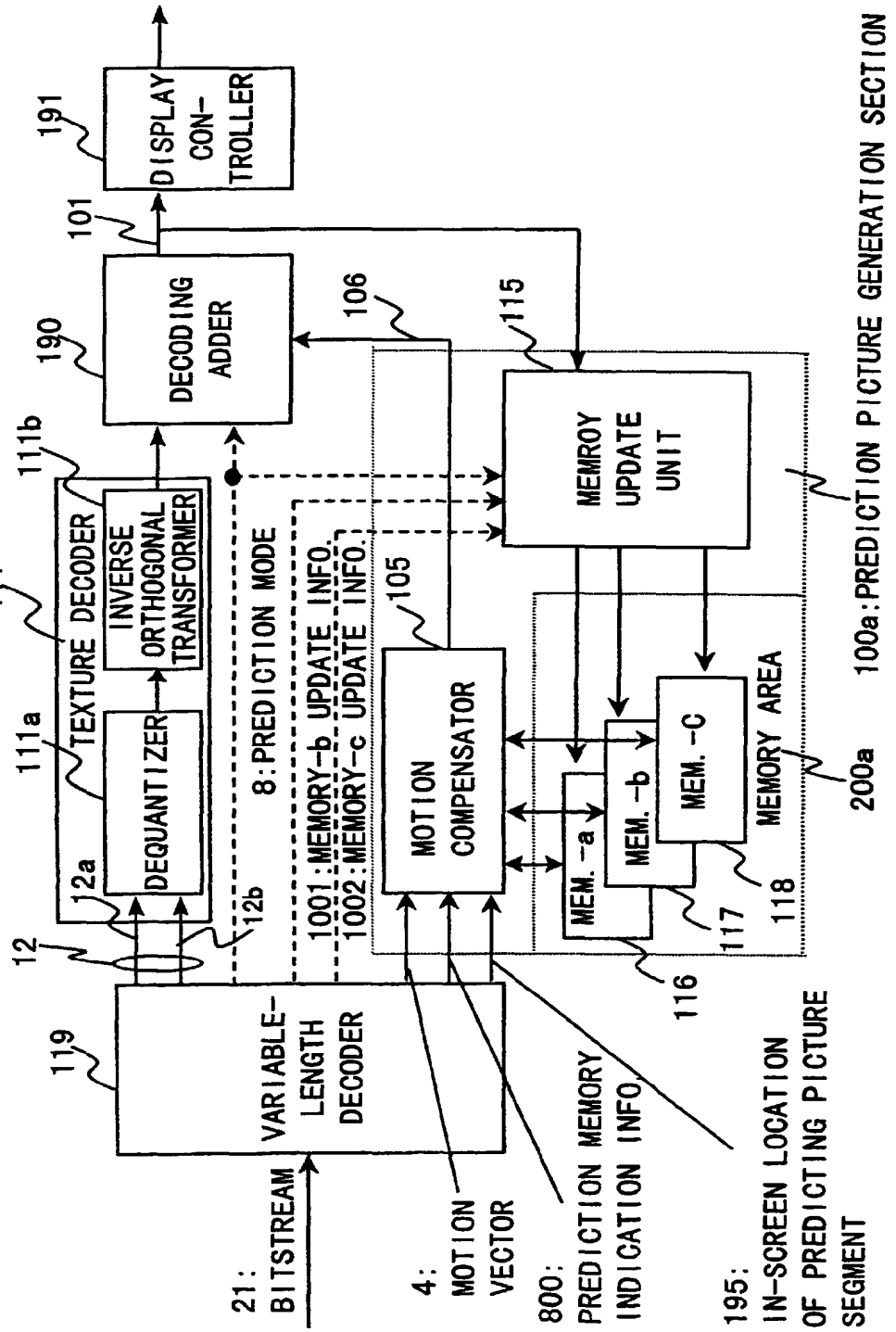

191
DISPLAY CON-TROLLER
101
190
DECODING ADDER
106
115
MEMROY UPDATE UNIT
111b
111
TEXTURE DECODER
INVERSE ORTHOGONAL TRANSFORMER
111a
DEQUANTIZER
12a
12b
8: PREDICTION MODE
1001: MEMORY-b UPDATE INFO.
1002: MEMORY-c UPDATE INFO.
105
MOTION COMPENSATOR
MEM. -a
MEM. -b
MEM. -C
MEMORY AREA
200a
100a: PREDICTION PICTURE GENERATION SECTION
12
119
VARIABLE-LENGTH DECODER
116
117
118
21: BITSTREAM
4: MOTION VECTOR
800: PREDICTION MEMORY INDICATION INFO.
195: IN-SCREEN LOCATION OF PREDICTING PICTURE SEGMENT

Fig. 25

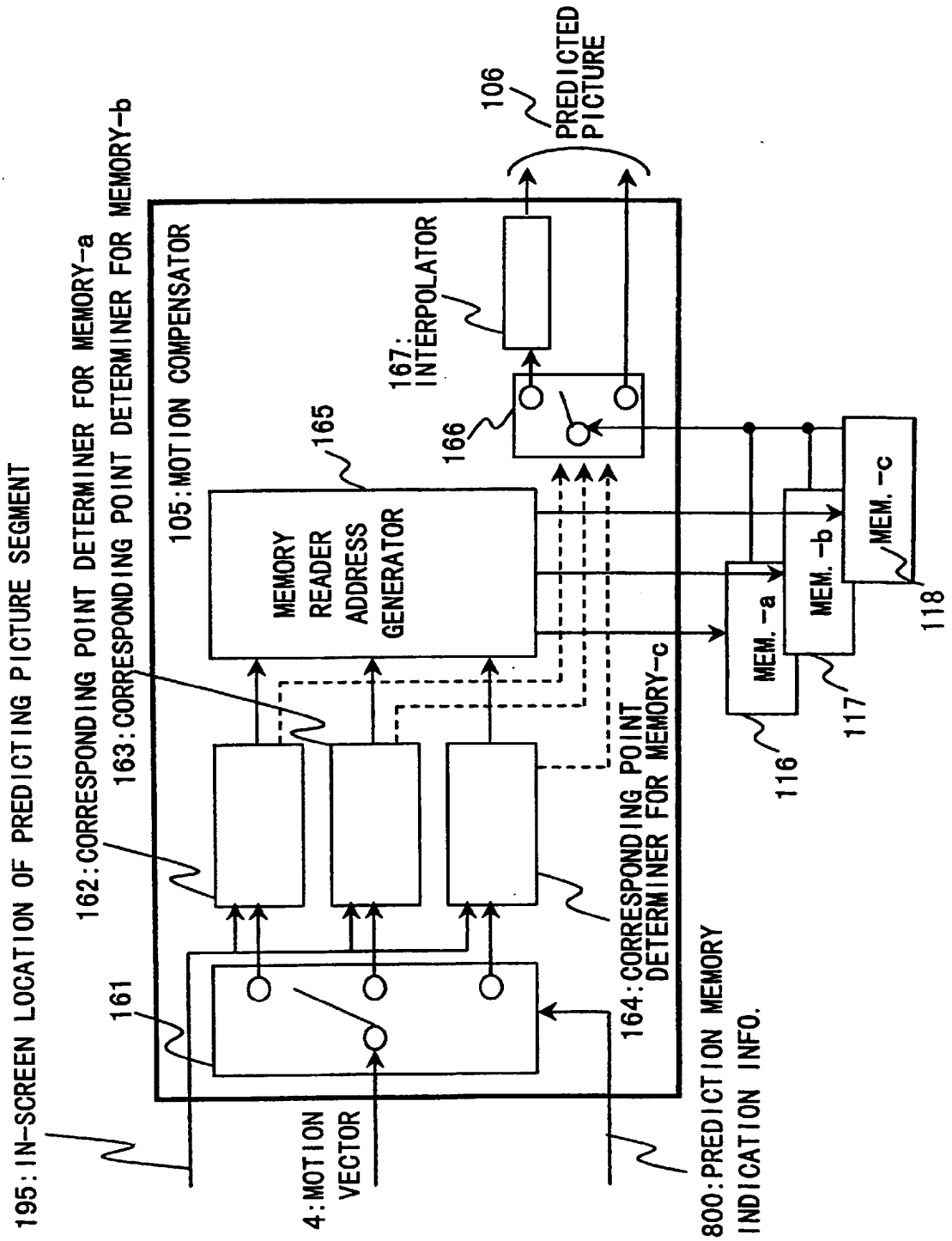

195: IN-SCREEN LOCATION OF PREDICTING PICTURE SEGMENT
162: CORRESPONDING POINT DETERMINER FOR MEMORY-a
163: CORRESPONDING POINT DETERMINER FOR MEMORY-b
105: MOTION COMPENSATOR
106
PREDICTED PICTURE
167: INTERPOLATOR
166
165
MEMORY READER ADDRESS GENERATOR
161
4: MOTION VECTOR
164: CORRESPONDING POINT DETERMINER FOR MEMORY-c
800: PREDICTION MEMORY INDICATION INFO.
MEM. -a
MEM. -b
MEM. -c
116
117
118

Fig. 26

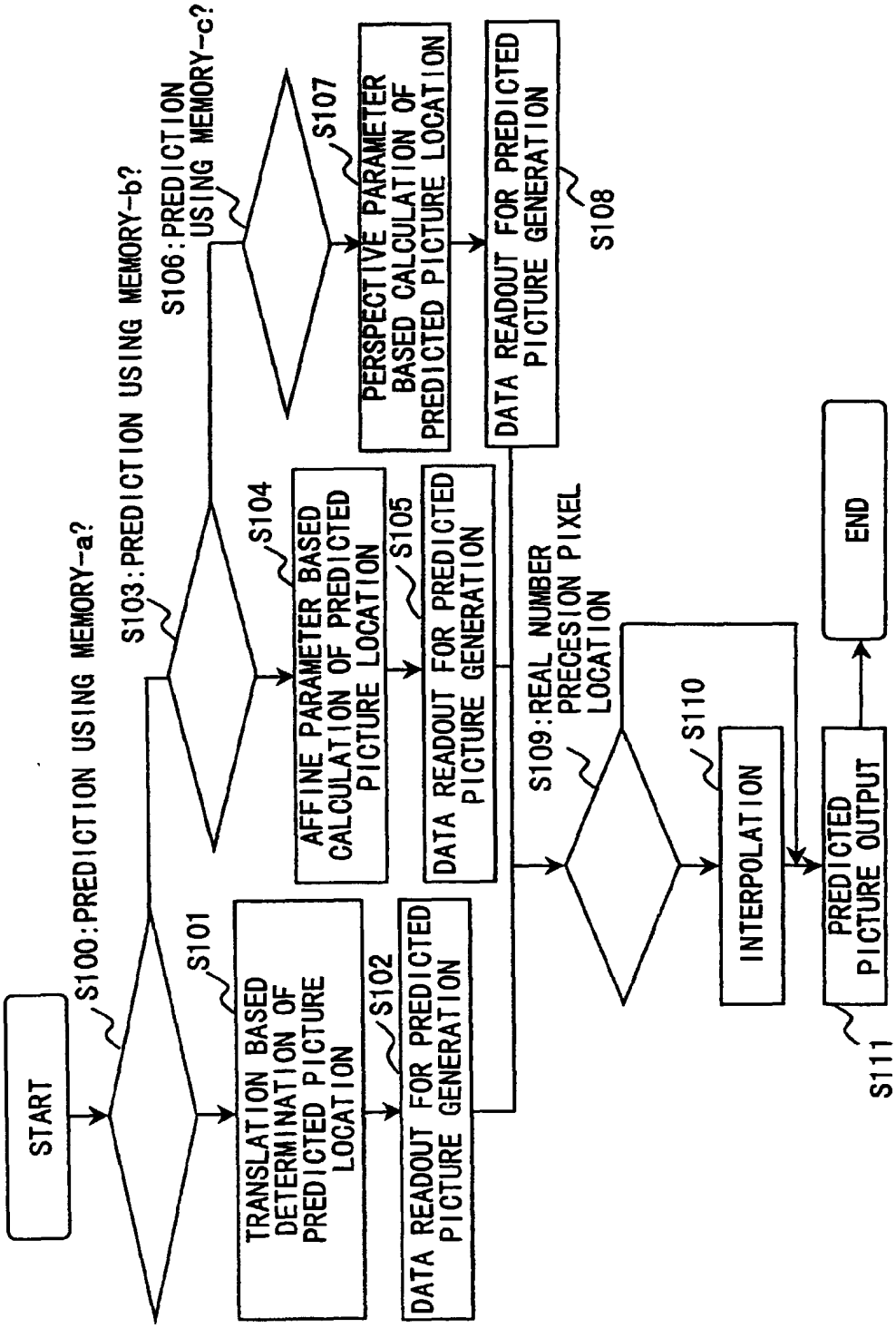

START
S100:PREDICTION USING MEMORY-a?
S101
TRANSLATION BASED DETERMINATION OF PREDICTED PICTURE LOCATION
S102
DATA READOUT FOR PREDICTED PICTURE GENERATION
S103:PREDICTION USING MEMORY-b?
S104
AFFINE PARAMETER BASED CALCULATION OF PREDICTED PICTURE LOCATION
S105
DATA READOUT FOR PREDICTED PICTURE GENERATION
S106:PREDICTION USING MEMORY-c?
S107
PERSPECTIVE PARAMETER BASED CALCULATION OF PREDICTED PICTURE LOCATION
S108
DATA READOUT FOR PREDICTED PICTURE GENERATION
S109:REAL NUMBER PRECESION PIXEL LOCATION
S110
INTERPOLATION
PREDICTED PICTURE OUTPUT
S111
END

Fig. 27

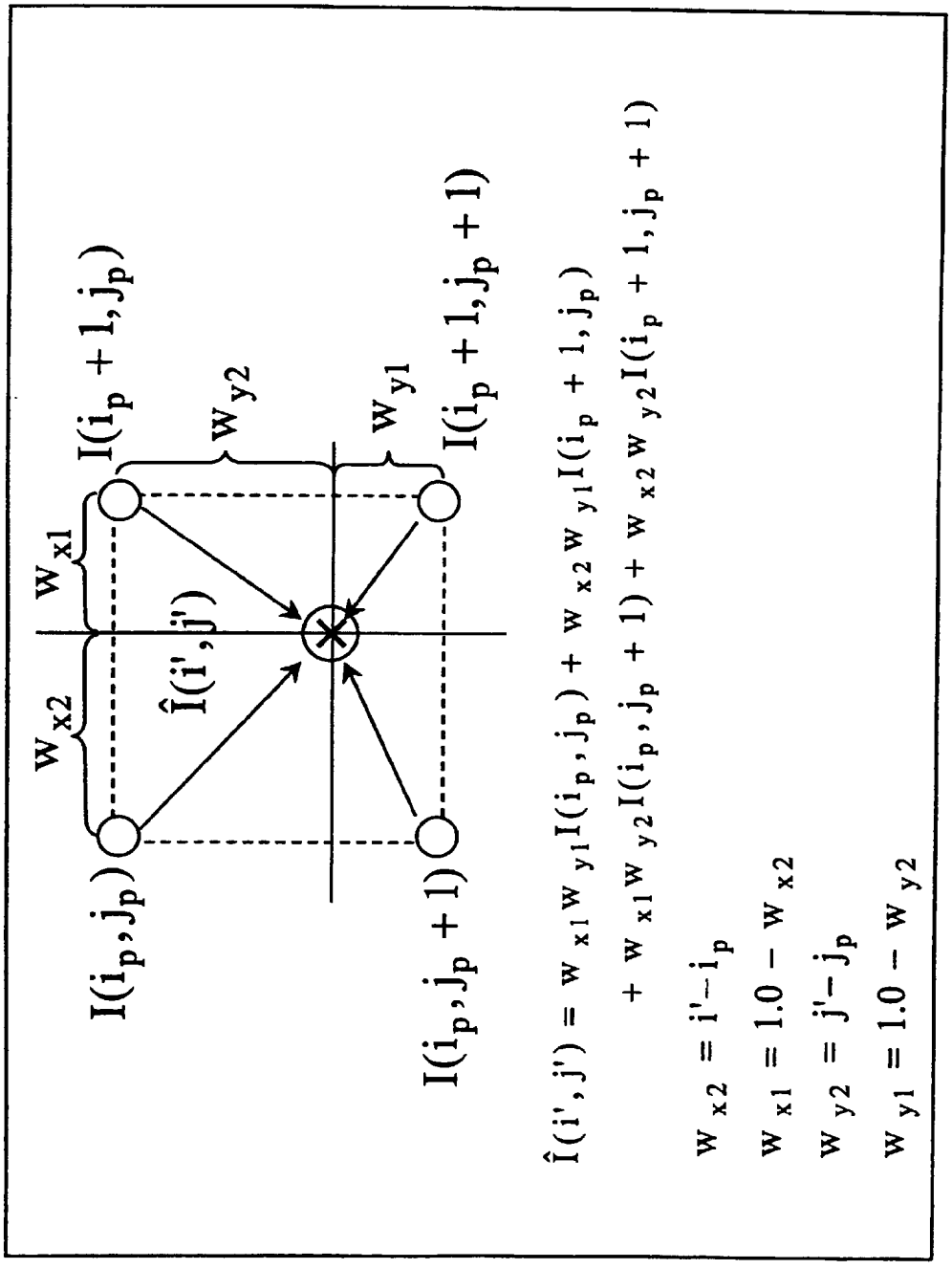

$I(i_p, j_p)$
$I(i_p + 1, j_p)$
$I(i_p, j_p + 1)$
$I(i_p + 1, j_p + 1)$
$\hat{I}(i', j')$
w_{x1}
w_{x2}
w_{y1}
w_{y2}
$\hat{I}(i', j') = w_{x1} w_{y1} I(i_p, j_p) + w_{x2} w_{y1} I(i_p + 1, j_p)$
$+ w_{x1} w_{y2} I(i_p, j_p + 1) + w_{x2} w_{y2} I(i_p + 1, j_p + 1)$
$w_{x2} = i' - i_p$
$w_{x1} = 1.0 - w_{x2}$
$w_{y2} = j' - j_p$
$w_{y1} = 1.0 - w_{y2}$

Fig. 28

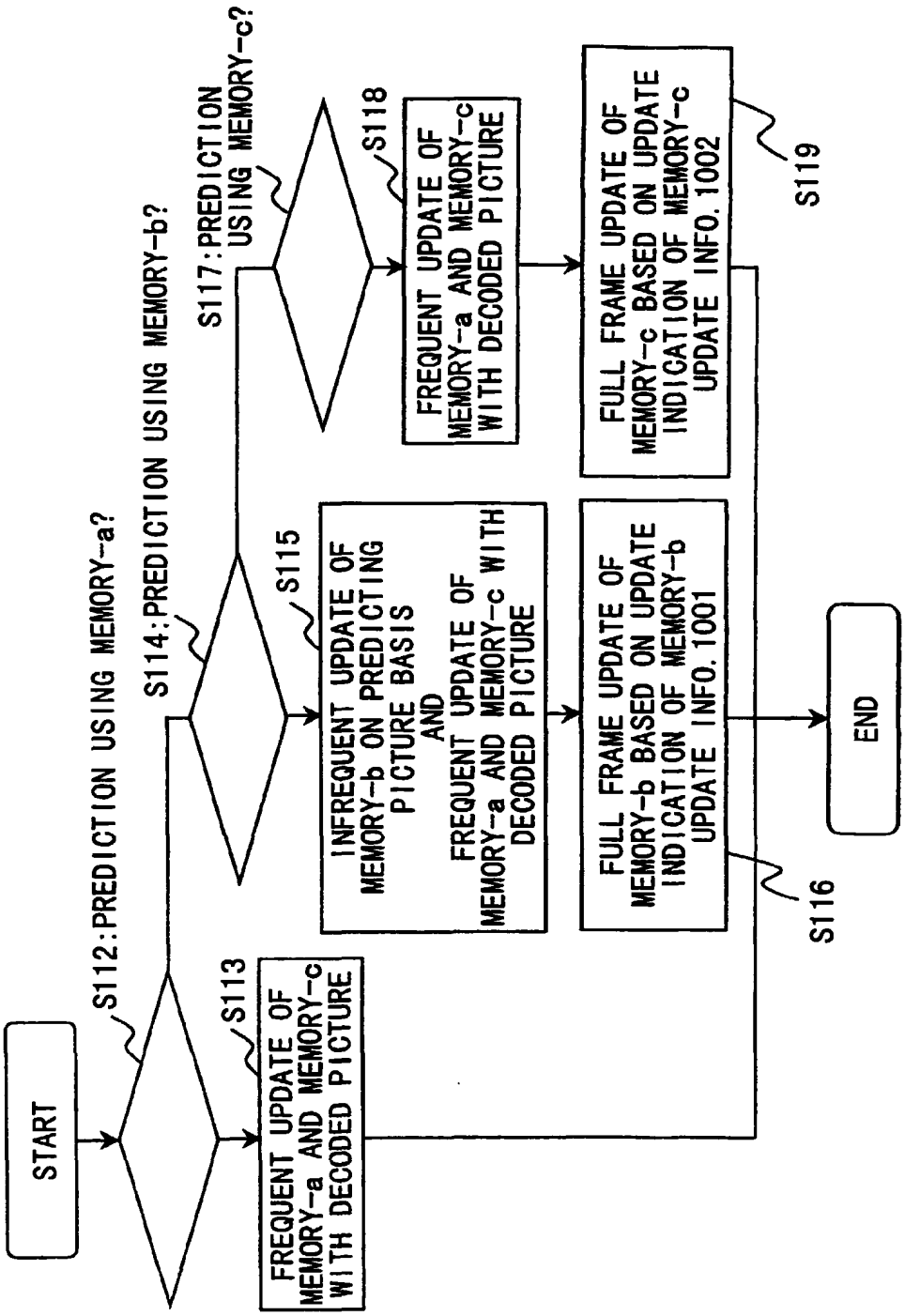

START
S112:PREDICTION USING MEMORY-a?
S113
FREQUENT UPDATE OF MEMORY-a AND MEMORY-c WITH DECODED PICTURE
S114:PREDICTION USING MEMORY-b?
S115
INFREQUENT UPDATE OF MEMORY-b ON PREDICTING PICTURE BASIS AND FREQUENT UPDATE OF MEMORY-a AND MEMORY-c WITH DECODED PICTURE
S116
FULL FRAME UPDATE OF MEMORY-b BASED ON UPDATE INDICATION OF MEMORY-b UPDATE INFO.1001
S117:PREDICTION USING MEMORY-c?
S118
FREQUENT UPDATE OF MEMORY-a AND MEMORY-c WITH DECODED PICTURE
S119
FULL FRAME UPDATE OF MEMORY-c BASED ON UPDATE INDICATION OF MEMORY-c UPDATE INFO.1002
END

Fig.29

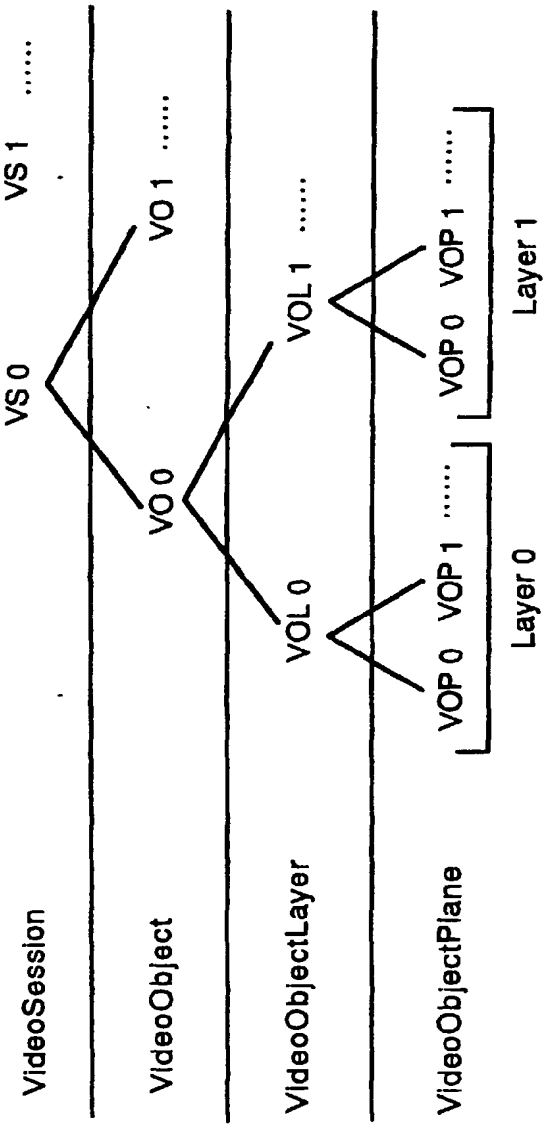

VideoSession
VideoObject
VideoObjectLayer
VideoObjectPlane
VS 0
VS 1 .....
VO 0
VO 1 .....
VOL 0
VOL 1 .....
VOP 0 VOP 1 .....
VOP 0 VOP 1 .....
Layer 0
Layer 1

Fig.30

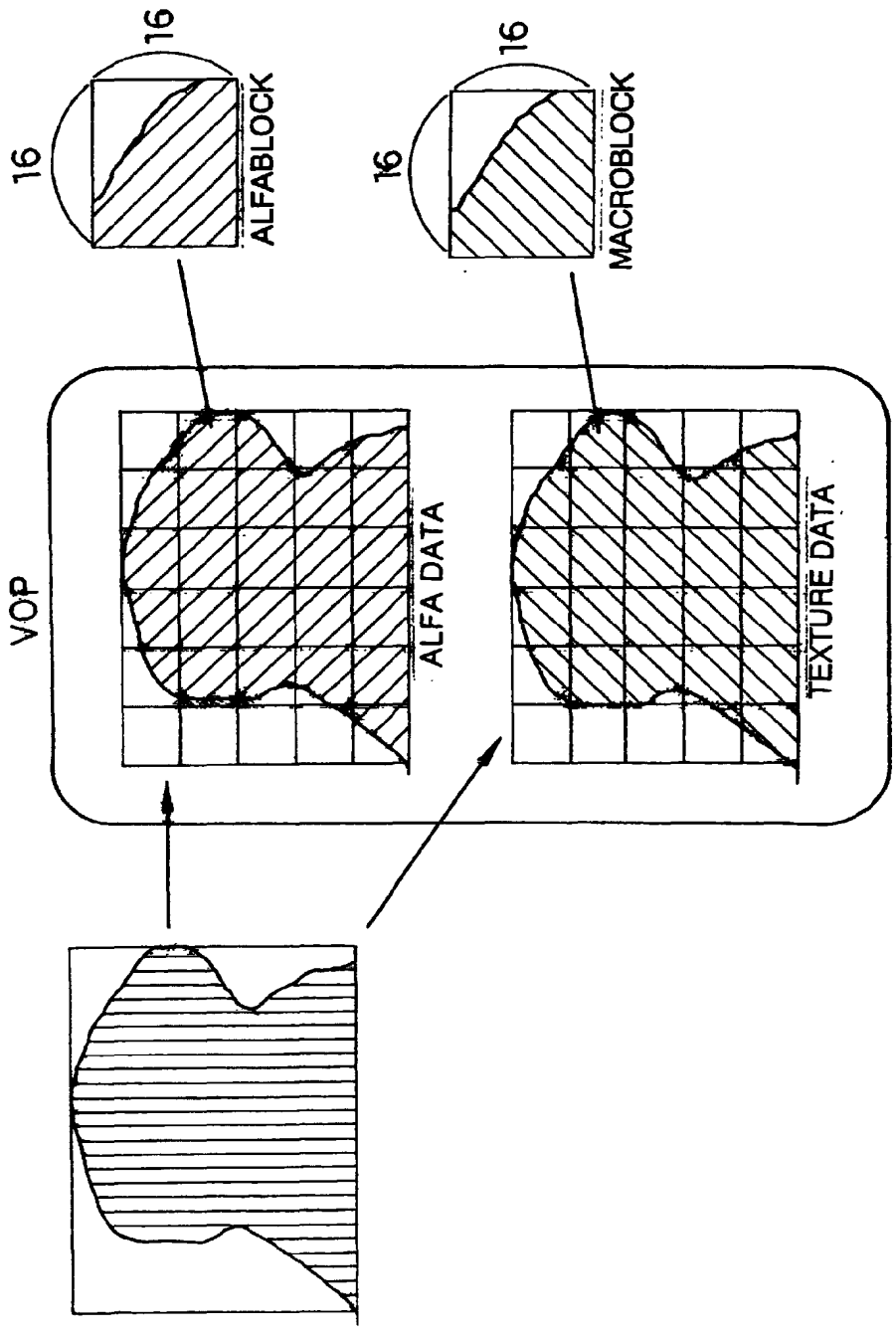


Fig.31

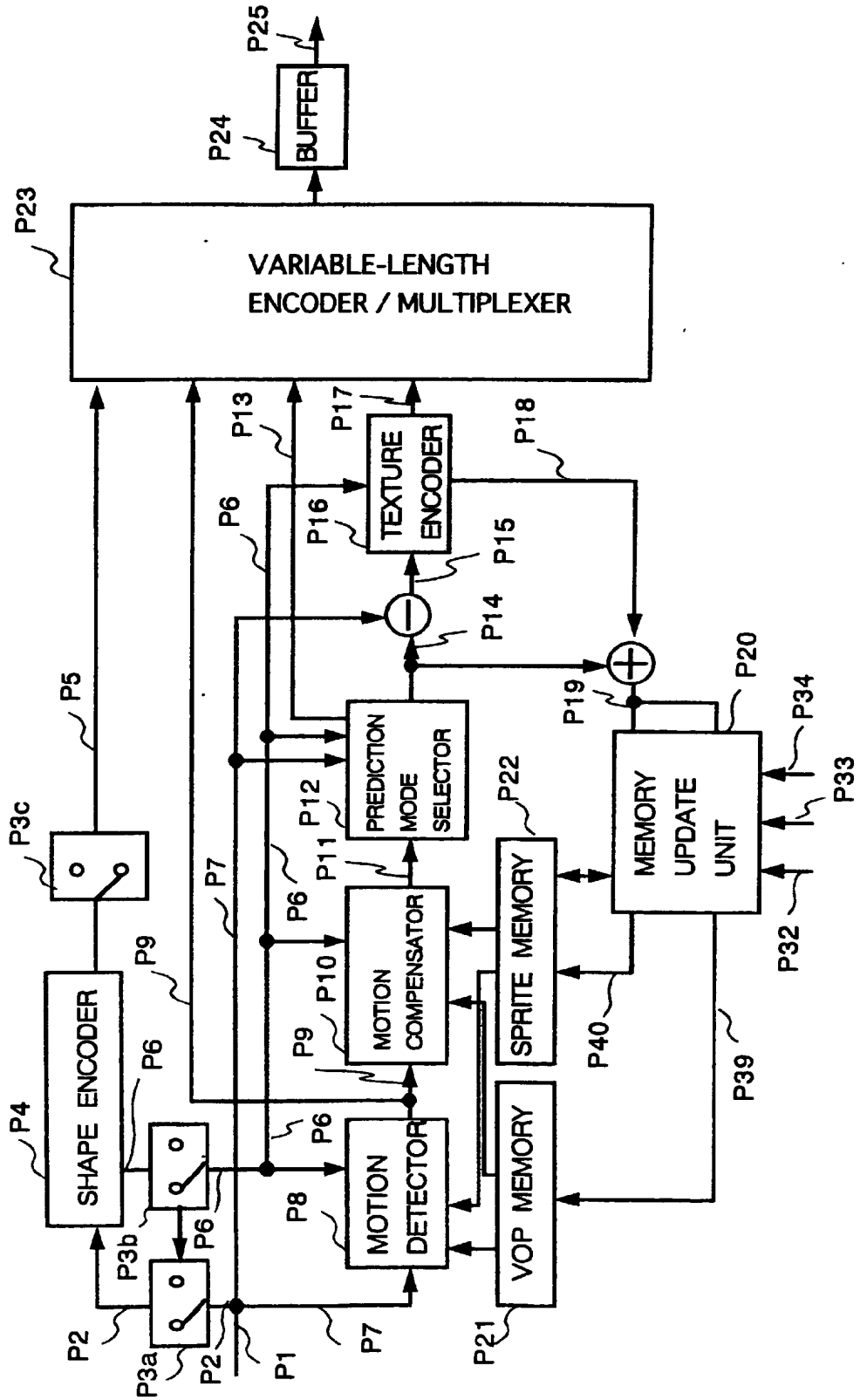

VARIABLE-LENGTH ENCODER / MULTIPLEXER
BUFFER
TEXTURE ENCODER
PREDICTION MODE SELECTOR
MOTION COMPENSATOR
MOTION DETECTOR
SHAPE ENCODER
SPRITE MEMORY
VOP MEMORY
MEMORY UPDATE UNIT
P1
P2
P3a
P3b
P3c
P4
P5
P6
P7
P8
P9
P10
P11
P12
P13
P14
P15
P16
P17
P18
P19
P20
P21
P22
P23
P24
P25
P32
P33
P34
P39
P40

Fig.32

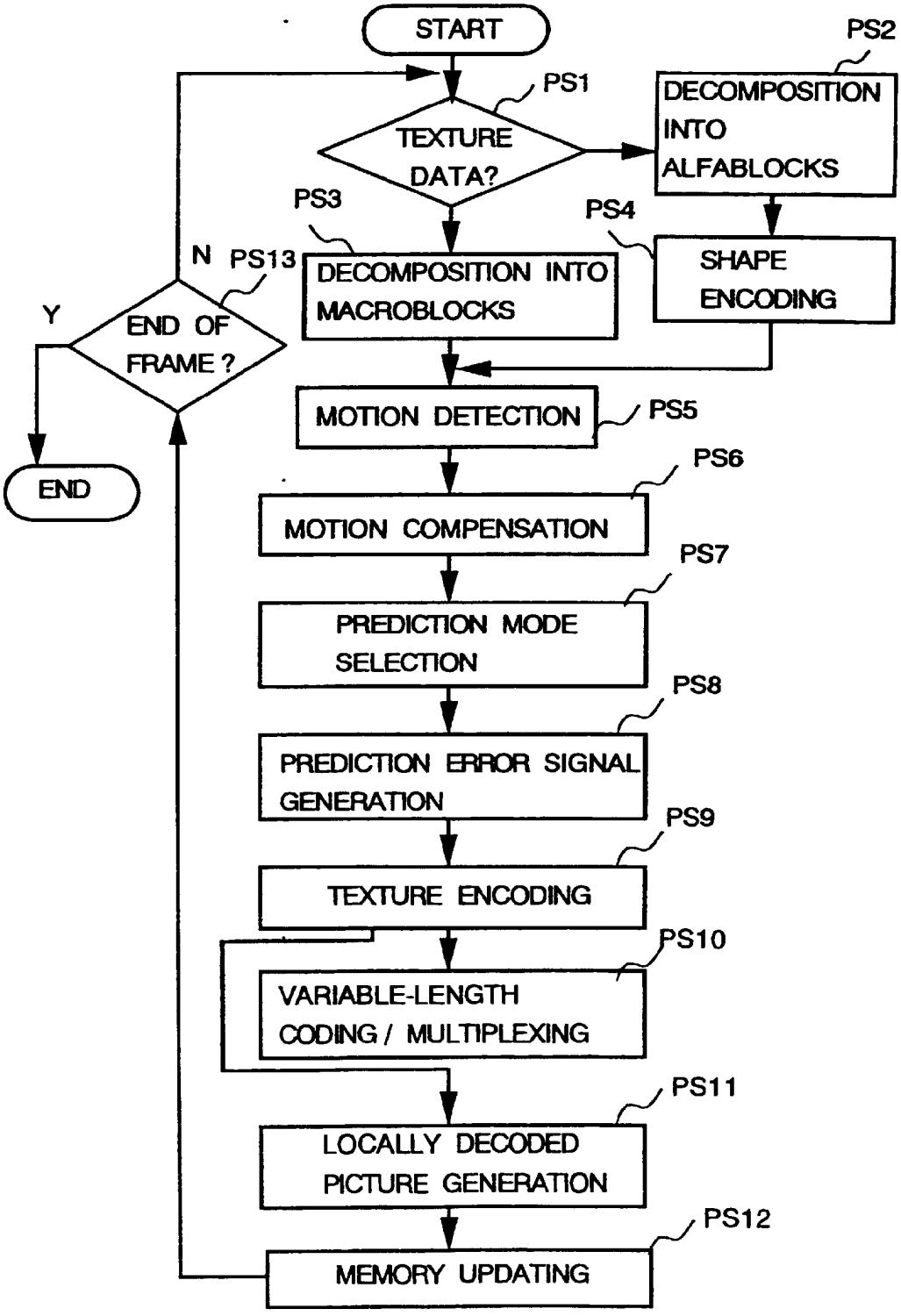

START
PS1
TEXTURE DATA?
PS2
DECOMPOSITION INTO ALFABLOCKS
PS3
DECOMPOSITION INTO MACROBLOCKS
PS4
SHAPE ENCODING
N
PS13
END OF FRAME ?
Y
END
MOTION DETECTION
PS5
MOTION COMPENSATION
PS6
PREDICTION MODE SELECTION
PS7
PREDICTION ERROR SIGNAL GENERATION
PS8
TEXTURE ENCODING
PS9
VARIABLE-LENGTH CODING / MULTIPLEXING
PS10
LOCALLY DECODED PICTURE GENERATION
PS11
MEMORY UPDATING
PS12

Fig.33

VOP TYPE	NO PREDICTION	PAST VOP BASED PREDICTION	PAST AND FUTURE VOP BASED PREDICTION	SPRITE BASED PREDICTION
I - VOP	O	X	X	X
P - VOP	O	O	X	X
B - VOP	O	O	O	X
SPRITE - VOP	O	O	X	O

Fig.34

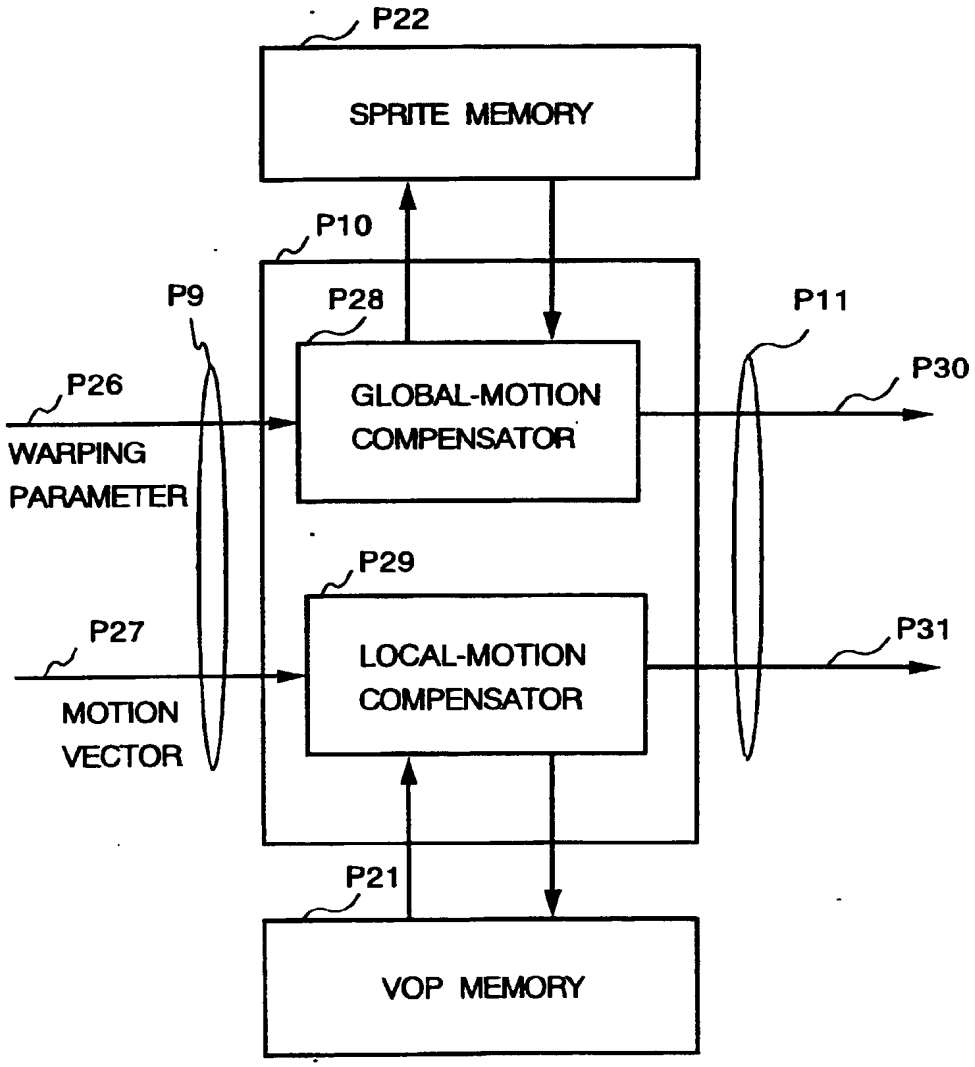

P22
SPRITE MEMORY
P10
P9
P28
P11
P30
P26
WARPING
PARAMETER
GLOBAL-MOTION
COMPENSATOR
P29
P27
P31
LOCAL-MOTION
COMPENSATOR
MOTION
VECTOR
P21
VOP MEMORY

Fig.35

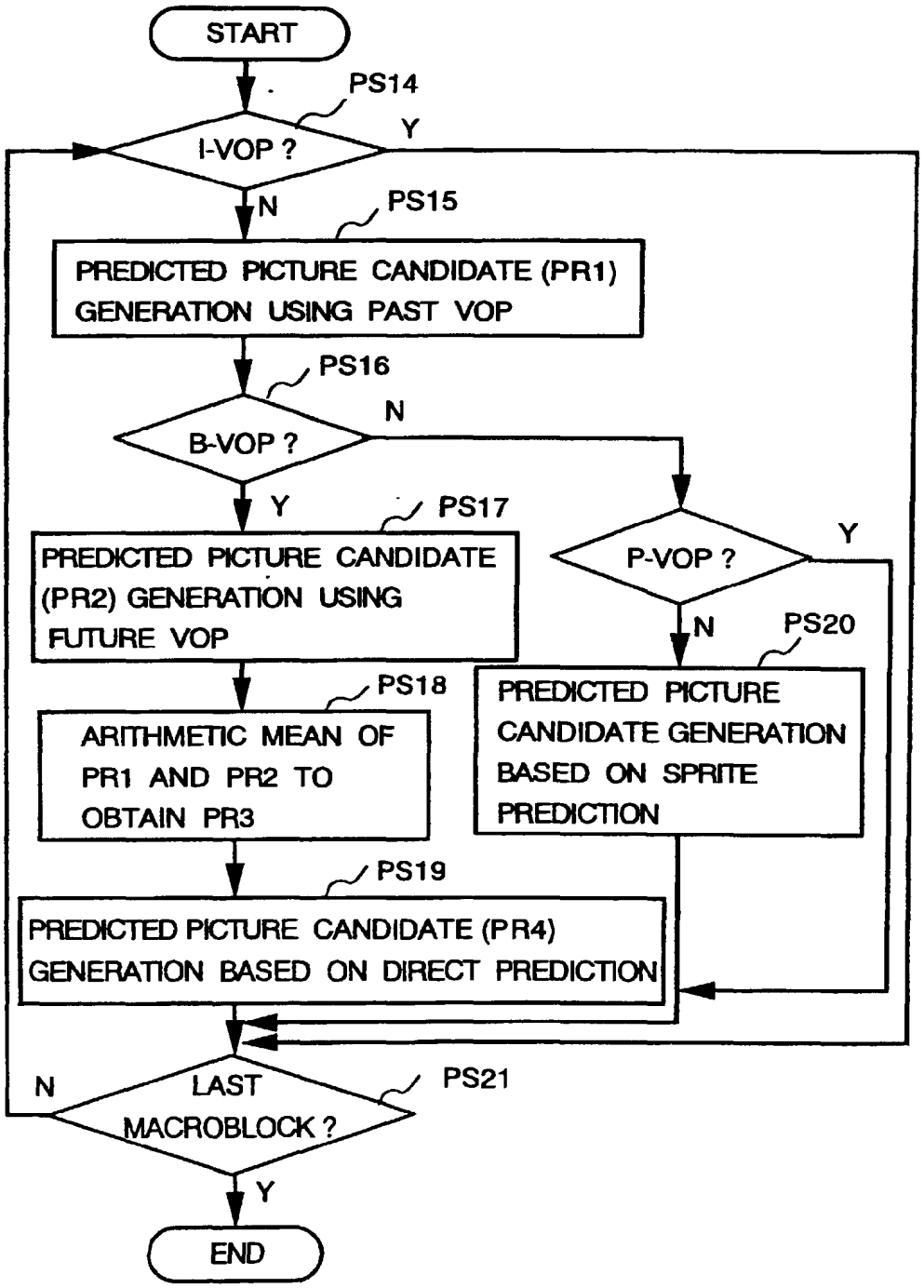

START
PS14
I-VOP ?
Y
N
PS15
PREDICTED PICTURE CANDIDATE (PR1) GENERATION USING PAST VOP
PS16
B-VOP ?
N
Y
PS17
PREDICTED PICTURE CANDIDATE (PR2) GENERATION USING FUTURE VOP
P-VOP ?
Y
N
PS20
PREDICTED PICTURE CANDIDATE GENERATION BASED ON SPRITE PREDICTION
PS18
ARITHMETIC MEAN OF PR1 AND PR2 TO OBTAIN PR3
PS19
PREDICTED PICTURE CANDIDATE (PR4) GENERATION BASED ON DIRECT PREDICTION
N
LAST MACROBLOCK ?
PS21
Y
END

Fig.36

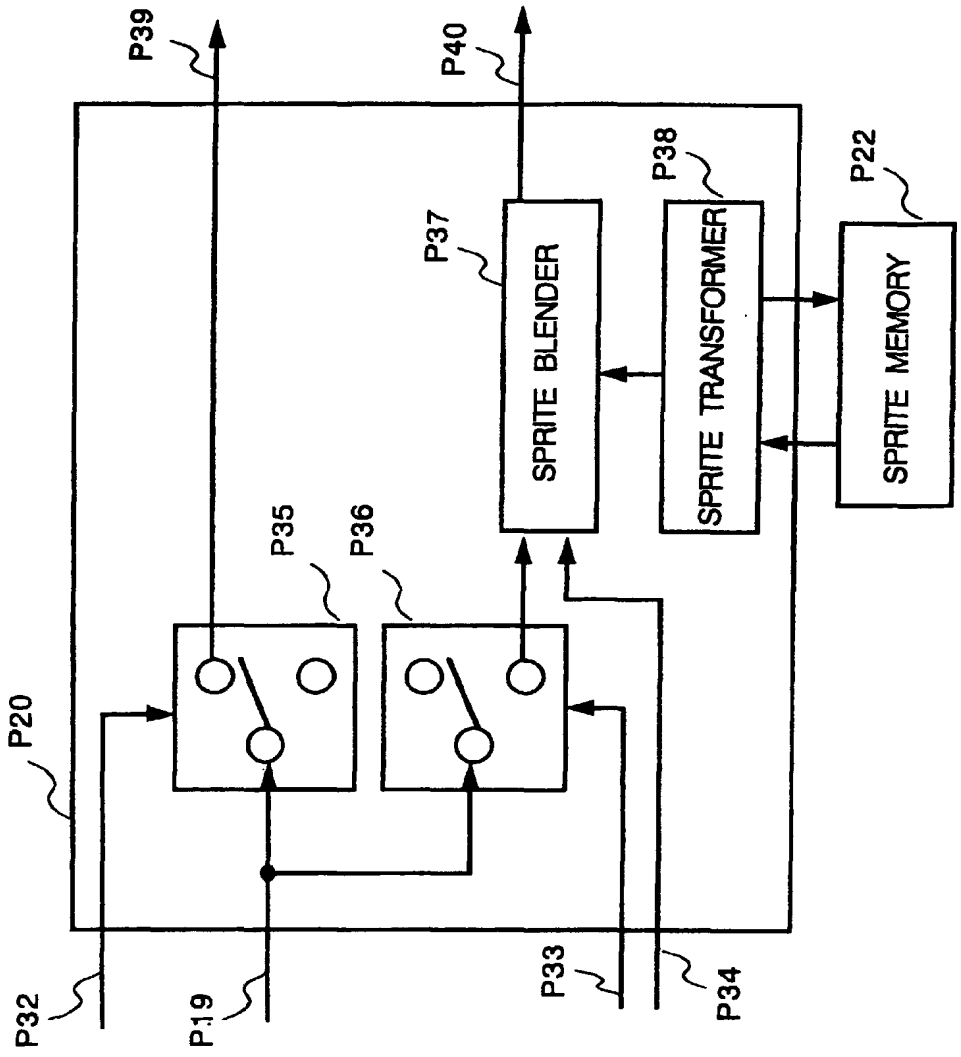
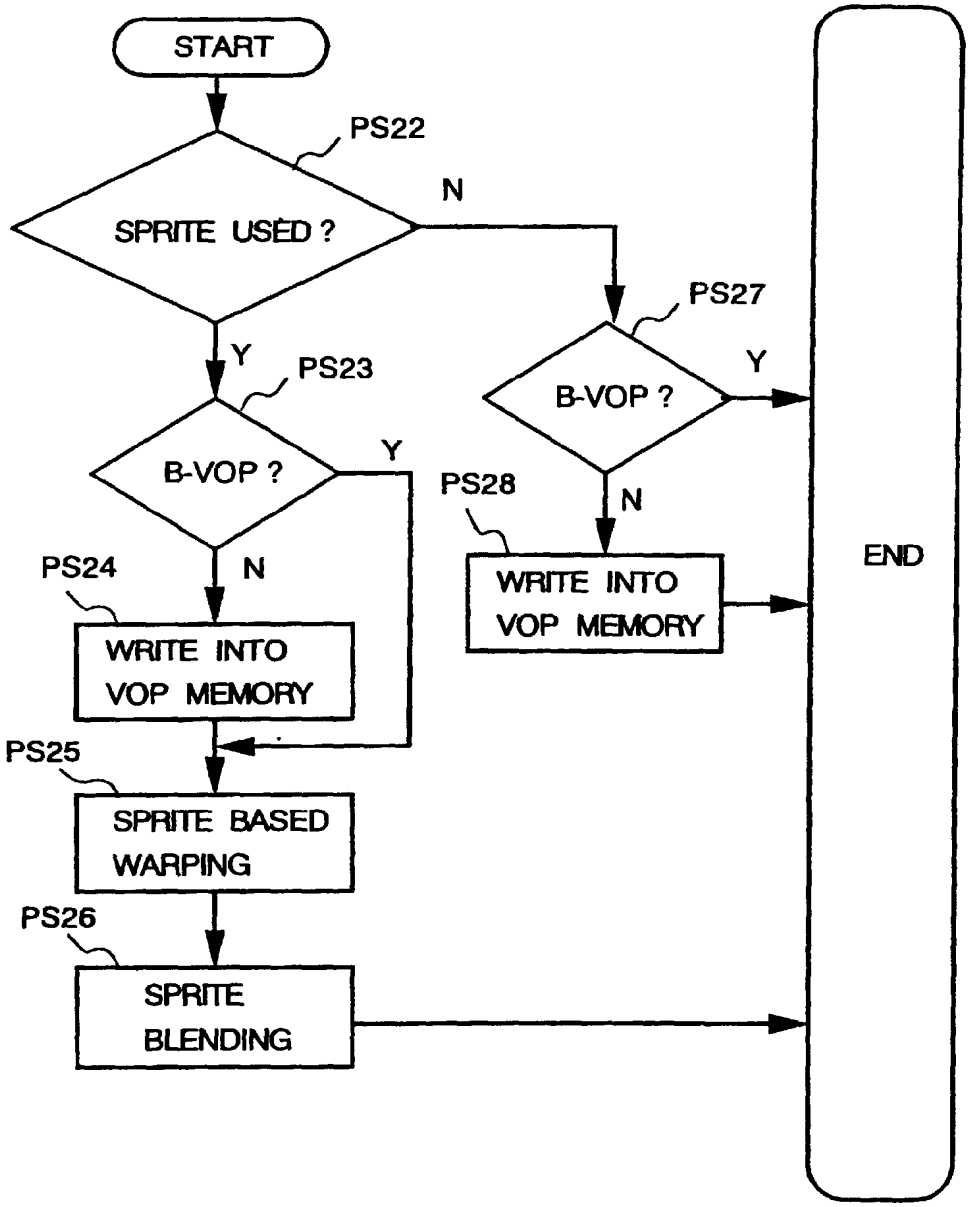

P39
P40
P38
P22
P37
SPRITE BLENDER
SPRITE TRANSFORMER
SPRITE MEMORY
P35
P36
P20
P32
P19
P33
P34

Fig.37


START
PS22
SPRITE USED ?
N
Y
PS23
B-VOP ?
Y
PS24
N
WRITE INTO VOP MEMORY
PS25
SPRITE BASED WARPING
PS26
SPRITE BLENDING
PS27
B-VOP ?
Y
PS28
N
WRITE INTO VOP MEMORY
END

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP98/00232

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[6] H04N7/32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl[6] H04N7/24-H04N7/68

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1957-1998
Kokai Jitsuyo Shinan Koho 1975-1998

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP, 8-172631, A (Matsushita Electric Industrial Co., Ltd.), July 2, 1996 (02. 07. 96) & EP, 719049, A2	1-3, 5, 9, 17-19, 21
Y		14-16
A		4, 6-8, 10-13, 20
Y	Hironao Nyozawa "Efficiency of Encoding in Conformity with MPEG-4 (in Japanese)", Preprints of 1996 Winter Meeting of Image Media Section, The Institute of Television Engineers of Japan, December 4, 1996 (04. 12. 96) (Tokyo) pages 39 to 44, particularly page 44, left column, lines 2 to 13	14-16
A		1-13, 17-21

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
April 21, 1998 (21. 04. 98)	May 12, 1998 (12. 05. 98)
Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer
Facsimile No.	Telephone No.

Form PCT/ISA/210 (second sheet) (July 1992)

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Diep, N.
Serial No.:	10/306,942	Group Art Unit:	2613
Filed:	November 29, 2002	Docket No.:	TAPA.004PA
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service with sufficient postage, as first class mail, in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on May 5, 2006.

By _____
Tracey M. Dotter

TRANSMITTAL

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

We are submitting herewith the following:

- ☒ Transmittal Sheet containing certificate under 37 CFR 1.8.
- ☒ Response to Restriction Requirement (11 pages).
- ☒ Petition for Extension of Time
- ☒ Information Disclosure Statement, PTO Form 1449, and reference.
- ☒ 1 Return Postcard

Authorization is given to charge/credit Deposit Account No. 50-3581 (TAPA.004PA) for any deficiencies/overpayments associated with this filing.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: May 5, 2006

By: _____
Erin M. Nichols
Reg. No. 57,125

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Diep, N.
Serial No.:	10/306,942	Group Art Unit:	2613
Filed:	November 29, 2002	Docket No.:	TAPA.004PA
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service with sufficient postage, as first class mail, in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on May 5, 2006.

By Tracey M. Dotter
Tracey M. Dotter

PETITION FOR EXTENSION OF TIME

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.136, it is respectfully requested that a one-month extension of time be granted in which to file the attached Restriction Response. Please charge Deposit Account No. 50-3581 (TAPA.004PA) in the amount $120.00 to cover the extension fee for a large entity. Authority is given to charge/credit necessary fees/overages to deposit account number 50-3581 (TAPA.004PA) as required by this filing.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700 (tel.)

By: E M Nichols
Name: Erin M. Nichols
Reg. No.: 57,125

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Diep, N.
Serial No.:	10/306,942	Group Art Unit:	2613
Filed:	November 29, 2002	Docket No.:	TAPA.004PA
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service with sufficient postage, as first class mail, in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on May 5, 2006.

By _____
Tracey M. Dotter

RESPONSE TO RESTRICTION REQUIREMENT AND PRELIMINARY AMENDMENT

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Examiner's requirement for restriction dated March 16, 2006, the claims of Group I (Claims 1-13 and 15-25) are provisionally elected *with traverse*. The Applicant respectfully requests reconsideration and withdrawal of the requirement for restriction, for reasons set forth below. The Applicant respectfully requests reconsideration and allowance of all pending claims.

Claims 26 and 27 have been added. Authorization is given to charge Deposit Account No. 50-3581 (TAPA.004PA) in the amount of $300.00 for the two additional claims added by this filing. Authorization is also given to charge/credit Deposit Account No. 50-3581 (TAPA.004PA) for any deficiencies/overpayments.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 10 of this paper.

Listing of Claims:

1. (Original) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

forming into the video sequence information indicating which video frames belong to the second sub-sequence.

2. (Original) A method according to claim 1, further comprising

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

3. (Original) A method according to claim 2, further comprising

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from a group comprising

- a video frame of a higher scalability layer,
- a video frame of another sub-sequence in the same scalability layer.

4. (Original) A method according to claim 3, further comprising

determining the dependencies of the video frames of the second sub-sequence on the basis of at least a scalability layer identifier and a sub-sequence identifier.

5. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

6. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

7. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.

8. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame on the basis of the image number.

9. (Original) A method according to claim 7, further comprising

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. (Original) A method according to claim 7, further comprising

adding the identifier to the Supplemental Enhancement Information (SEI) data structure to be transmitted synchronously with the video sequence.

11. (Original) A method according to claim 1, wherein

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

12. (Original) A method according to claim 1, further comprising

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

13. (Original) A method according to claim 1, further comprising

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

14. (Original) An element in a streaming system, such as a streaming server or a network element in a telecommunications system, wherein said element is arranged to

transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission,

control the bit rate of a video sequence transmission in which a first sub-sequence of the video sequence comprises video frames of at least a first frame format and at least a second sub-sequence comprises video frames of at least a second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

the bit rate control being arranged to be carried out in at least one of the following ways:

- at least one sub-sequence is removed from the video sequence,
- at least one sub-sequence is added to the video sequence.

15. (Original) A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format; and to

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame

of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and to

form into the video sequence information indicating which video frames belong the second sub-sequence.

16. (Original) A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

decoding a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;

decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determining dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and

reconstructing at least part of the video sequence on the basis of the dependencies of the sub-sequences.

17. (Original) A method according to claim 16, wherein video frames are entered into a sliding buffer memory in connection with decoding, further comprising

decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;

configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

18. (Original) A method according to claim 17, further comprising

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

19. (Original) A method according to claim 16, further comprising

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

20. (Original) A method according to claim 16, further comprising

initiating the decoding from a random point in the video sequence;

determining an independently decodable sub-sequence that is next after the random point, and

setting the value of the image number of the first video frame in the sub-sequence at zero.

21. (Original) A method according to claim 16, further comprising

identifying at least partly temporally overlapping sub-sequence frames from the video sequence based on the picture order count information, and

outputting from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.

22. (Original) A video decoder for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video decoder is arranged to

decode a first sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the first frame format;

decode at least a second sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video

frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.

23. (Original) A computer program for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining data into the video sequence to determine which video frames belong to the second sub-sequence.

24. (Original) A computer program for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second

frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.

25. (Original) A video signal comprising a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video signal comprises

a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format;

at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

at least one data field determining the video frames that belong to the second sub-sequence.

26. (New) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format;

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

forming into the video sequence information indicating which video frames belong to the second sub-sequence, wherein subsequence information is different from picture type information.

27. (New) A method according to claim 26, wherein

a removal of the second subsequence from the bitstream does not prevent decoding of the bitstream correctly.

REMARKS/ARGUMENTS

The election/restriction requirement dated March 16, 2006, has been considered. The Examiner contends that the claims are drawn to two inventions, namely Group I (Claims 1-13 and 15-25) and Group II (Claim 14) and asserts that restriction to one of claim Groups I-II is required under 35 U.S.C. §121. The claims of **Group I (Claims 1-13 and 15-25)** are provisionally elected *with traverse*. The Applicant traverses the restriction requirement as it fails to satisfy the requirements for maintaining such a restriction requirement and accordingly requests reconsideration and withdrawal of the restriction requirement.

The election/restriction requirement asserts that the claims of Groups I and II are unrelated. It appears that the Examiner is asserting that the claims of the two Groups are independent based upon the reliance on §§802.01 and 806.06 of the MPEP. Specifically, §806.06 indicates that two inventions are independent if there is no disclosed relationship between the inventions, that is, they are unconnected in design, operation, and effect. The standard for showing that two inventions are independent is difficult to prove, for example, a specific process of molding would be independent from a molding apparatus so long as the apparatus cannot be used to practice the specific process. The Examiner's assertion that the claims of Group I are directed to forming a scalable, compressed video sequence, whereas the claim of Group II is directed to control of the bit rate of the compressed video packets for transmission, fails to identify how the two alleged inventions are disclosed as not capable of use together. The Examiner fails to provide any evidence to support the conclusion that the two alleged inventions are independent. Without any such evidence, the restriction requirement is improper. Accordingly, the Applicant requests that the restriction requirement be withdrawn.

New Claims 26 and 27 have been added. Claim 26 largely replicates Claim 1 and is believed to be directed to a similar invention if the above election/restriction requirement is maintained; therefore, the new claims are believed to be associated with the Examiner's asserted Group I.

CONCLUSION

In view of the above, the Applicant respectfully requests reconsideration and withdrawal of the requirement for restriction. If the Examiner would find it helpful to discuss this issue by telephone, the undersigned attorney of record invites the Examiner to contact the attorney of record.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

By: E. M. Nichols

Erin M. Nichols
Reg. No. 57,125

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Diep, N.
Serial No.:	10/306,942	Group Art Unit:	2621
Filed:	November 29, 2002	Docket No.:	TAPA.004PA

Title: GROUPING OF IMAGE FRAMES IN VIDEO CODING

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this Transmittal Letter and the papers, as described herein, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on May 31, 2007.

By: Rennae Johnson

AMENDMENT & RESPONSE UNDER 37 C.F.R. §1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office Action mailed March 13, 2007, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. §1.111.

Claims 1-24, 26 and 27 remain pending in the application. Claims 1, 2 and 26 have been amended. Claim 25 has been canceled, and no claims have been added. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 10.

AMENDMENTS TO THE CLAIMS:

This listing of claims replaces all prior versions, and listings, of claims in the application.

1. (Currently amended) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

encoding~~forming into~~ the video sequence as at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format; and

encoding~~forming into~~ the video sequence as at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

encoding~~forming~~ into the video sequence information indicating which video frames belong to the second sub-sequence.

2. (Currently amended) A method according to claim 1, further comprising

determining ~~into the video sequence~~ the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence; and

encoding said interdependency into the video sequence.

3. (Original) A method according to claim 2, further comprising

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from a group comprising

- a video frame of a higher scalability layer,
- a video frame of another sub-sequence in the same scalability layer.

4. (Original) A method according to claim 3, further comprising

determining the dependencies of the video frames of the second sub-sequence on the basis of at least a scalability layer identifier and a sub-sequence identifier.

5. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

6. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

7. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.

8. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame on the basis of the image number.

9. (Original) A method according to claim 7, further comprising

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. (Original) A method according to claim 7, further comprising

adding the identifier to the Supplemental Enhancement Information (SEI) data structure to be transmitted synchronously with the video sequence.

11. (Original) A method according to claim 1, wherein

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

12. (Original) A method according to claim 1, further comprising

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

13. (Original) A method according to claim 1, further comprising

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

14. (Withdrawn) An element in a streaming system, such as a streaming server or a network element in a telecommunications system, wherein said element is arranged to

transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission,

control the bit rate of a video sequence transmission in which a first sub-sequence of the video sequence comprises video frames of at least a first frame format and at least a second sub-sequence comprises video frames of at least a second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

the bit rate control being arranged to be carried out in at least one of the following ways:

- at least one sub-sequence is removed from the video sequence,

- at least one sub-sequence is added to the video sequence.

15. (Original) A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format; and to

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and to

form into the video sequence information indicating which video frames belong the second sub-sequence.

16. (Original) A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

decoding a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;

decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determining dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and

reconstructing at least part of the video sequence on the basis of the dependencies of the sub-sequences.

17. (Original) A method according to claim 16, wherein video frames are entered into a sliding buffer memory in connection with decoding, further comprising

decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;

configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

18. (Original) A method according to claim 17, further comprising

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

19. (Original) A method according to claim 16, further comprising

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

20. (Original) A method according to claim 16, further comprising

initiating the decoding from a random point in the video sequence;

determining an independently decodable sub-sequence that is next after the random point, and

setting the value of the image number of the first video frame in the sub-sequence at zero.

21. (Original) A method according to claim 16, further comprising

identifying at least partly temporally overlapping sub-sequence frames from the video sequence based on the picture order count information, and

outputting from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.

22. (Original) A video decoder for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video decoder is arranged to

decode a first sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the first frame format;

decode at least a second sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.

23. (Previously presented) A computer program product, stored on a computer readable medium and executable in a data processing device, for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining data into the video sequence to determine which video frames belong to the second sub-sequence.

24. (Previously presented) A computer program product, stored on a computer readable medium and executable in a data processing device, for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.

25. (Canceled)

26. (Currently amended) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

encoding~~forming into~~ the video sequence as at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format;

encoding~~forming into~~ the video sequence as at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

encoding~~forming~~ into the video sequence information indicating which video frames belong to the second sub-sequence, wherein subsequence information is different from picture type information.

27. (Previously presented) A method according to claim 26, wherein

a removal of the second subsequence from the bitstream does not prevent decoding of the bitstream correctly.

REMARKS/ARGUMENTS

The rejection presented in the Office Action dated March 13, 2007 (hereinafter Office Action) has been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

Claims 3-10, 12, 13 and 15-24 are not mentioned, or identified as being rejected, in the instant Office Action; therefore these claims are believed to be allowable. Applicant notes that MPEP §707.07(d) indicates that where a claim is refused the word "reject" must be used and the statutory basis for any ground of rejection should be designated by an express reference in the opening sentence of each ground of rejection. As the Office Action does not indicate that these claims are rejected, Applicant accordingly requests that each of these claims be identified as being allowable. If this was not the Examiner's intention, Applicant requests clarification, an opportunity to respond, and that any future rejections comply with MPEP §707.07(d).

With respect to the rejection of Claim 25, Claim 25 has been canceled thereby rendering the rejection moot. Applicant accordingly requests that the rejection be withdrawn.

Regarding the §101 rejection of Claims 1, 2, 11, 26 and 27, Applicant respectfully traverses because the claimed invention is directed to statutory subject matter. Independent Claims 1 and 26 are directed to processes for coding video frames, and expressly recite at least one specific utility in that the coding is "for the purpose of forming a scalable, compressed video sequence." It has not been asserted or shown that the claimed coding constitutes a judicial exception to the statutory categories of subject matter such as an abstract idea. Moreover, MPEP §2106(IV)(C)(2) indicates that a claimed invention is directed to a practical application of a judicial exception when it produces a useful, concrete and tangible result, and MPEP §2106(IV)(C)(2)(2) further indicates that the focus is on whether the final result achieved by the claimed invention is useful, tangible, and concrete. While the Examiner has previously acknowledged that the claims have useful and concrete results, Applicant notes that MPEP §2106(IV)(C)(2)(2)(b) does not limit a "real-world result" to a physical object. The claimed subject matter produces as a result, an encoded

video sequence of video frames having information indicating which frames belong to a second sub-sequence. The resulting encoded video sequence is used for the practical application of forming a scalable, compressed video sequence. As it has not been shown that the claimed scalable, compressed video sequence is not a practical application of encoded video frames, Applicant maintains that the claimed invention is directed to statutory subject matter and requests that the rejection be withdrawn.

Applicant has also amended Claims 1, 2 and 26 to more clearly characterize that the video sequence is encoded. These changes merely incorporate subject matter that was implicit in the previous claim language and therefore the changes do not introduce new matter. Each of these claims is believed to be patentable for the reasons discussed above.

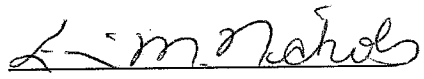
Applicant also notes that as of August of 2006, the Interim Guidelines referred to at page two of the Office Action were incorporated into MPEP §2106.

Authorization is given to charge Deposit Account No. 50-3581 (TAPA.004PA) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the undersigned attorney of record invites the Examiner to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: May 31, 2007

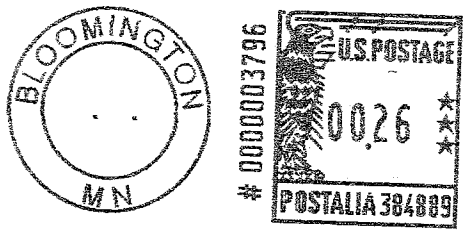
By: 

Erin M. Nichols
Reg. No. 57,125

Receipt is hereby acknowledged for the following in the U.S. Patent and Trademark Office:
Applicant: HANNUKSELA
Serial No.: 10/306,942
Docket No.: TAPA.004PA
Date of Deposit: May 31, 2007

☒ Transmittal Sheet containing certificate under 37 CFR 1.8.
☒ Office Action Amendment and Response (11 pages)
☒ 1 Return Postcard

Authorization is given to charge/credit Deposit Account No. 50-3581 (TAPA.004PA) for any deficiencies/overpayments.



HOLLINGSWORTH & FUNK, LLC
8009 34TH AVENUE SOUTH, SUITE 125
MINNEAPOLIS MN 55425

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Diep, N.
Serial No.:	10/306,942	Group Art Unit:	2621
Filed:	November 29, 2002	Docket No.:	TAPA.004PA
Confirmation No.:	3642	Customer No.:	75385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on July 9, 2009.

By: /Rennae Johnson /
Rennae Johnson

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office Action mailed April 9, 2009, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.111.

Claims 1-13, 15-24, 26, and 27 remain pending in the application. Claims 1, 16, and 26 have been amended. No claims have been added or canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 10.

AMENDMENTS TO THE CLAIMS:

This listing of claims replaces all prior versions of claims in the application.

1. (Currently amended) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

encoding, at an encoder, the video sequence as at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format; and

encoding, at the encoder, the video sequence as at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

encoding, at the encoder, into the video sequence information indicating which video frames belong to the second sub-sequence.

2. (Previously presented) A method according to claim 1, further comprising

determining the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence; and

encoding said interdependency into the video sequence.

3. (Original) A method according to claim 2, further comprising

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from a group comprising

- a video frame of a higher scalability layer,

- a video frame of another sub-sequence in the same scalability layer.

4. (Original) A method according to claim 3, further comprising

determining the dependencies of the video frames of the second sub-sequence on the basis of at least a scalability layer identifier and a sub-sequence identifier.

5. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

6. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

7. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.

8. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame on the basis of the image number.

9. (Original) A method according to claim 7, further comprising

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. (Original) A method according to claim 7, further comprising

adding the identifier to the Supplemental Enhancement Information (SEI) data structure to be transmitted synchronously with the video sequence.

11. (Original) A method according to claim 1, wherein

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

12. (Original) A method according to claim 1, further comprising

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

13. (Original) A method according to claim 1, further comprising

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

14. (Canceled)

15. (Original) A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format; and to

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and to

form into the video sequence information indicating which video frames belong the second sub-sequence.

16. (Currently amended) A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

decoding, at a decoder, a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;

decoding, at the decoder, at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determining, at the decoder, dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and

reconstructing, at the decoder, at least part of the video sequence on the basis of the dependencies of the sub-sequences.

17. (Original) A method according to claim 16, wherein video frames are entered into a sliding buffer memory in connection with decoding, further comprising

decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;

configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

18. (Original) A method according to claim 17, further comprising

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

19. (Original) A method according to claim 16, further comprising

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

20. (Original) A method according to claim 16, further comprising

initiating the decoding from a random point in the video sequence;

determining an independently decodable sub-sequence that is next after the random point, and

setting the value of the image number of the first video frame in the sub-sequence at zero.

21. (Original) A method according to claim 16, further comprising

identifying at least partly temporally overlapping sub-sequence frames from the video sequence based on the picture order count information, and

outputting from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.

22. (Original) A video decoder for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video decoder is arranged to

decode a first sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the first frame format;

decode at least a second sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video

frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.

23. (Previously presented) A computer program product, stored on a computer readable medium and executable in a data processing device, for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining data into the video sequence to determine which video frames belong to the second sub-sequence.

24. (Previously presented) A computer program product, stored on a computer readable medium and executable in a data processing device, for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other

video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.

25. (Canceled)

26. (Currently amended) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

encoding, at an encoder, the video sequence as at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format;

encoding, at the encoder, the video sequence as at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

encoding, at the encoder, into the video sequence information indicating which video frames belong to the second sub-sequence, wherein subsequence information is different from picture type information.

27. (Previously presented) A method according to claim 26, wherein

a removal of the second subsequence from the bitstream does not prevent decoding of the bitstream correctly.

REMARKS/ARGUMENTS

The rejection presented in the Office Action dated April 9, 2009, (hereinafter Office Action) has been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

Applicant appreciates the indication of allowance for Claims 15 and 22-24.

With particular respect to the § 101 rejection of Claims 1-13, 16-21, 26, and 27, independent Claims 1, 16, and 26 have been amended to indicate that the claimed methods are performed at either an encoder or a decoder thereby tying the methods to another statutory category of subject matter as evidenced by the allowance of Claims 15 and 22. Although the subject matter of these changes was implicitly present in the claims, further support for these changes may be found, for example, in original Claims 15 and 22; therefore, these changes do not introduce new matter. Since the changes are believed to overcome the rejection, Applicant requests that the rejection be withdrawn and each of the pending claims be allowed.

Authorization is given to charge Deposit Account No. 50-3581 (TAPA.004PA) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the undersigned attorney of record invites the Examiner to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: July 9, 2009

By: /Erin M. Nichols/

Erin M. Nichols
Reg. No. 57,125

Electronic Acknowledgement Receipt

EFS ID:	5670390
Application Number:	10306942
International Application Number:	
Confirmation Number:	3642
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TAPA.004PA
Receipt Date:	09-JUL-2009
Filing Date:	29-NOV-2002
Time Stamp:	13:29:49
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		OAR-TAPA004PA.pdf	81922 f573e012a4eb7e76e8454f5436dbd1a187c3524b	yes	10

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment/Req. Reconsideration-After Non-Final Reject	1	1
	Claims	2	9
	Applicant Arguments/Remarks Made in an Amendment	10	10

Warnings:	
Information:	
Total Files Size (in bytes):	81922

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Diep, N.
Serial No.:	10/306,942	Group Art Unit:	2621
Filed:	November 29, 2002	Docket No.:	**TAPA.004PA** Old: KOLS.021PA

Title: GROUPING OF IMAGE FRAMES IN VIDEO CODING

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this Transmittal Letter and the papers, as described herein, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on November 9, 2006.

By: Rennae Johnson

AMENDMENT & RESPONSE UNDER 37 C.F.R. §1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office Action mailed August 10, 2006, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. §1.111.

Claims 1-27 remain pending in the application. Claims 23 and 24 have been amended. No claims have been added or canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 11.

AMENDMENTS TO THE CLAIMS:

This listing of claims replaces all prior versions, and listings, of claims in the application.

1. (Original) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

forming into the video sequence information indicating which video frames belong to the second sub-sequence.

2. (Original) A method according to claim 1, further comprising

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

3. (Original) A method according to claim 2, further comprising

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from a group comprising

- a video frame of a higher scalability layer,

- a video frame of another sub-sequence in the same scalability layer.

4. (Original) A method according to claim 3, further comprising

determining the dependencies of the video frames of the second sub-sequence on the basis of at least a scalability layer identifier and a sub-sequence identifier.

5. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

6. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

7. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.

8. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame on the basis of the image number.

9. (Original) A method according to claim 7, further comprising

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. (Original) A method according to claim 7, further comprising

adding the identifier to the Supplemental Enhancement Information (SEI) data structure to be transmitted synchronously with the video sequence.

11. (Original) A method according to claim 1, wherein

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

12. (Original) A method according to claim 1, further comprising

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

13. (Original) A method according to claim 1, further comprising

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

14. (Withdrawn) An element in a streaming system, such as a streaming server or a network element in a telecommunications system, wherein said element is arranged to

transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission,

control the bit rate of a video sequence transmission in which a first sub-sequence of the video sequence comprises video frames of at least a first frame format and at least a second sub-sequence comprises video frames of at least a second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

the bit rate control being arranged to be carried out in at least one of the following ways:

- at least one sub-sequence is removed from the video sequence,
- at least one sub-sequence is added to the video sequence.

15. (Original) A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format; and to

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and to

form into the video sequence information indicating which video frames belong the second sub-sequence.

16. (Original) A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

decoding a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;

decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determining dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and

reconstructing at least part of the video sequence on the basis of the dependencies of the sub-sequences.

17. (Original) A method according to claim 16, wherein video frames are entered into a sliding buffer memory in connection with decoding, further comprising

decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;

configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

18. (Original) A method according to claim 17, further comprising

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

19. (Original) A method according to claim 16, further comprising

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

20. (Original) A method according to claim 16, further comprising

initiating the decoding from a random point in the video sequence;

determining an independently decodable sub-sequence that is next after the random point, and

setting the value of the image number of the first video frame in the sub-sequence at zero.

21. (Original) A method according to claim 16, further comprising

identifying at least partly temporally overlapping sub-sequence frames from the video sequence based on the picture order count information, and

outputting from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.

22. (Original) A video decoder for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video decoder is arranged to

decode a first sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the first frame format;

decode at least a second sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.

23. (Currently amended) A computer program product, stored on a computer readable medium and executable in a data processing device, for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining data into the video sequence to determine which video frames belong to the second sub-sequence.

24. (Currently amended) A computer program product, stored on a computer readable medium and executable in a data processing device, for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.

25. (Original) A video signal comprising a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video signal comprises

a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format;

at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

at least one data field determining the video frames that belong to the second sub-sequence.

26. (Previously presented) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format;

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

forming into the video sequence information indicating which video frames belong to the second sub-sequence, wherein subsequence information is different from picture type information.

27. (Previously presented) A method according to claim 26, wherein

a removal of the second subsequence from the bitstream does not prevent decoding of the bitstream correctly.

REMARKS/ARGUMENTS

The rejections presented in the Office Action dated August 10, 2006 (hereinafter Office Action) have been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

While the Examiner has indicated that the restriction requirement has been made final, Applicant maintains the traversal of the requirement and reserves the right to petition for reversal of the requirement.

Applicant appreciates the indication of allowability for claims 17-18 and 20.

Regarding the rejection of Claims 1, 2, 11, 23, 24, 26 and 27 under 35 U.S.C. §101 as allegedly not being directed to statutory subject matter, Applicant respectfully traverses. First, the Examiner asserts that Claims 1, 2, 11, 26 and 25 do not indicate what to do with formed sub-sequences. Applicant fails to recognize how such an assertion relates to a rejection based on allegedly non-statutory subject matter and is unsure as to the basis for such an assertion. The claims clearly teach that "at least part of a first sub-sequence" and "at least a second sub-sequence" are formed "into the video sequence." *See, e.g.*, Claim 1.

Second, the Examiner asserts that Claims 1, 2, 11, 26 and 25 have useful and concrete results, but lack tangible results and therefore do not provide a practical application. Applicant fails to recognize how claimed subject matter may be directed to concrete results without being directed to tangible results, especially since "the courts have yet to define the terms useful, concrete, and tangible in the context of the practical application requirement." *See, e.g.*, MPEP §2106(II)(A). Notwithstanding, the claims are directed to a process for forming a scalable compressed video sequence, which includes coding video frames having video data. Such a manipulation of video data is a practical application in compliance with MPEP §2106(IV)(B)(2)(b). Thus, Applicant submits that Claims 1, 2, 11, 26 and 25 are directed to statutory subject matter and requests that the rejection be withdrawn. If the Examiner were to maintain the rejection, Applicant requests further clarification and an opportunity to respond.

With respect to Claims 23 and 24, Applicant has amended the claims to be directed to a computer program product, stored on a computer readable medium and executable in a data processing device. Such a computer program product constitutes statutory subject matter as it is directed to a practical application of forming a scalable compressed video sequence. *See, e.g.*, MPEP §2106(IV)(B)(2)(b). Applicant accordingly submits that Claims 23 and 24 are directed to statutory subject matter and requests that the rejection be withdrawn.

With respect to the prior art (§§102(e) and 103(a)) rejections based on U.S. Patent No. 6,614,936 to Wu *et al.* (hereinafter "Wu"), Applicant respectfully traverses because Wu does not teach or suggest each of the claimed limitations. For example, Wu does not teach at least forming into a video sequence, information indicating which video frames belong to a second sub-sequence, as claimed. In an attempt to show correspondence to such limitations, the Examiner asserts that information indicating that the second through third frames (asserted as corresponding to a second sub-sequence and shown in Fig. 11) belong to a first layer is transmitted to a decoder. However, the Examiner has not identified where Wu teaches that such information is formed into a video sequence or provided any other evidence in support of such assertion. Without a presentation of correspondence to each of the claimed limitations, the prior art rejections are improper.

With particular respect to the §102(e) rejection, Applicant notes that to anticipate a claim, the asserted reference must teach every element of the claim. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The identical invention must be shown in as complete detail as is contained in the patent claim; *i.e.* every element of the claimed invention must be literally present, arranged as in the claim. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Therefore, all claim elements, and their limitations, must be found in the prior art reference to maintain the rejection based on 35 U.S.C. §102. Applicant respectfully submits that Wu does not teach every element of

independent Claims 1, 15, 16 and 22-26 in the requisite detail and therefore fails to anticipate Claims 1-9, 11-13, 15-16, 19 and 22-27.

Even if Wu teaches transmitting information indicating that the second through third frames belong to a first layer, such teaching fails to correspond to the claimed limitations. For example, the fact that the relied-upon second through third frames belong to a first layer does not indicate that they belong to a second sub-sequence. The Examiner has not identified how the second through third frames are formed into a video sequence and at least part of the second through third frames was formed by coding using a second frame format. As multiple sub-sequences may be present within a layer (*see, e.g.*, Applicant's Fig. 2), the Examiner has not shown that the second through third frames correspond to the claimed second sub-sequence. Also, the Examiner has not shown that information indicating where the second through third frames belong is formed into a video sequence. The Examiner has not shown that Wu teaches each of these claim limitations in as complete detail as contained in the claims to maintain a §102 rejection. Applicant accordingly requests that the rejection be withdrawn.

Dependent Claims 2-9, 11-13, 19 and 27 depend from independent Claims 1, 16 and 26, respectively, and also stand rejected as being anticipated by Wu. While Applicant does not acquiesce with the particular rejections to these dependent claims, including any assertions concerning inherency, these rejections are moot in view of the remarks made above in connection with independent Claims 1, 16 and 26. These dependent claims include all of the limitations of independent Claims 1, 16 and 26 and any intervening claims, and recite additional features which further distinguish these claims from the cited references. Therefore, the rejection of dependent Claims 2-9, 11-13, 19 and 27 is improper and should be withdrawn.

With particular respect to the rejection of dependent Claim 7, Applicant respectfully traverses because Wu does not teach each of the claimed limitations. The Examiner has not identified where Wu teaches any determination of unique identifiers for each video frame. Moreover, the Examiner's assertion that the first frame of the base layer is known as base layer, first sub-sequence, first frame, is entirely unsupported since Wu does not use the term

"sub-sequence" anywhere in its teachings. Without a presentation of correspondence to each of the claimed limitations, the §102(e) rejection is improper and should not be maintained.

With particular respect to the §103(a) rejection of dependent Claim 10, Applicant respectfully traverses because the asserted modification of Wu would not correspond to the claimed limitations and is unsupported. As discussed above in connection with dependent Claim 7, Wu does not teach determining a unique identifier, thus Wu must also not teach adding the identifier to Supplemental Enhancement Information. It would appear that the Examiner is impermissibly taking official notice that the elements of Claim 10, in view of Wu, constitute facts outside of the record which are capable of instant and unquestionable demonstration as being well known or obvious to one skilled in the art. Applicant respectfully disagrees and requests that the Examiner produce a reference that supports the Examiner's assertion that Claim 10 is obvious.

In determining the differences between the prior art and the claims, the question under 35 U.S.C. §103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious. Taking official notice of the above-discussed "facts" disregards the requirement of analyzing Applicant's claimed subject matter "as a whole." Applicant respectfully reiterates the legal tenet that facts so noticed should not comprise the principle evidence upon which a rejection is based. *See*, MPEP §2144.03.

It is to be understood that Applicant does not acquiesce to the Examiner's characterization of the asserted art or Applicant's claimed subject matter, nor of the Examiner's application of the asserted art or combinations thereof to Applicant's claimed subject matter. Moreover, Applicant does not acquiesce to any explicit or implicit statements or conclusions by the Examiner concerning what would have been obvious to one of ordinary skill in the art, obvious design choices, alternative equivalent arrangements, common knowledge at the time of Applicant's invention, officially noticed facts, and the like. Applicant respectfully submits that a detailed discussion of each of the Examiner's rejections beyond that provided above is not necessary, in view of the clear absence of

teaching and suggestion of various features recited in Applicant's pending claims and a lack of evidence of motivation to modify the asserted teachings. Applicant, however, reserves the right to address in detail the Examiner's characterizations, conclusions, and rejections in future prosecution.

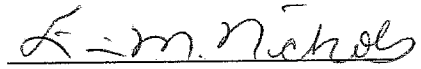
The Examiner has not shown that, nor does, Wu teach or suggest each of the claimed limitations. Thus, each of the prior art rejections are improper and should not be maintained. Applicant accordingly requests that each of the rejections be withdrawn.

Authorization is given to charge Deposit Account No. 50-3581 (TAPA.004PA) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the undersigned attorney of record invites the Examiner to contact her at the number below to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: November 9, 2006

By: 

Erin M. Nichols
Reg. No. 57,125

Receipt is hereby acknowledged for the following in the U.S. Patent and Trademark Office:
Applicant: HANNUKSELA
Serial No.: 10/306,942
Docket No.: TAPA.004PA
Date of Deposit: November 9, 2006

☒ Office Action Response (15 pages); with certificate under 37 CFR 1.8.

☒ 1 Return Postcard

HOLLINGSWORTH & FUNK, LLC
8009 34TH AVENUE SOUTH #125
MINNEAPOLIS MN 55425

© 2005 USPS recycled

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	KOLS.021PA	3642

7590 03/16/2006

Hollingsworth & Funk, LLC
Suite 125
8009 34th Avenue South
Minneapolis, MN 55425

EXAMINER	
DIEP, NHON THANH	
ART UNIT	**PAPER NUMBER**
2613	

DATE MAILED: 03/16/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/306,942	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	Nhon T. Diep	2613	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 1 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-25 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) _____ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☒ Claim(s) 1-25 are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Election/Restrictions

1. Restriction to one of the following inventions is required under 35 U.S.C. 121:

   I. Claims 1-13, 15-25, drawn to a method of coding video frame for forming a scalable, compressed video sequence, classified in class 375, subclass 240.12.

   II. Claim 14, drawn to a method of transmitting compressed video sequence and to control the transferred bit rate, classified in class 375, subclass 240.16.

The inventions are distinct, each from the other because of the following reasons:

Inventions I and II are unrelated. Inventions are unrelated if it can be shown that they are not disclosed as capable of use together and they have different designs, modes of operation, and effects (MPEP § 802.01 and § 806.06). In the instant case, the different inventions of group I requires the forming of a scalable, compressed video sequence whereas, the invention of group II relates to a control of the bit rate of the compressed video packets for transmission.

Because these inventions are independent or distinct for the reasons given above and have acquired a separate status in the art in view of their different classification, restriction for examination purposes as indicated is proper.

Because these inventions are independent or distinct for the reasons given above and the inventions require a different field of search (see MPEP § 808.02), restriction for examination purposes as indicated is proper.

Because these inventions are independent or distinct for the reasons given above and have acquired a separate status in the art because of their recognized divergent subject matter, restriction for examination purposes as indicated is proper.

2. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nhon T. Diep whose telephone number is 571-272-7328. The examiner can normally be reached on m-f.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mehrdad Dastouri can be reached on 571-272-7418. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

ND
3/11/2006

NHON DIEP
PRIMARY EXAMINER

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	KOLS.021PA	3642

7590 03/13/2007

Hollingsworth & Funk, LLC
Suite 125
8009 34th Avenue South
Minneapolis, MN 55425

EXAMINER	
DIEP, NHON THANH	
ART UNIT	**PAPER NUMBER**
2621	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/13/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/306,942	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	Nhon T. Diep	2621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133)
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13 and 15-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,11 and 25-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 November 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *11/24/2006*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 1-2, 11, 25-27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Regarding claims 1-2, 11, 26-27: Claims 1-2, 11, 26-27 claim a method for coding video frames for the purpose of forming a scalable, compressed video sequence. Since the method claim of claims 1-2, 11, 26-27 do not have substantial practical application as required in the Interim Guidelines, in particular, the tangible requirement does require that the claims must recite more than a 101 judicial exception, in that the process claim must set forth a practical application of that 101 judicial exception to produce a real-world result. Regarding to claims 1-17, the claims are determined lacking tangible result since all the steps cited in the claims, only manipulate data, evaluating data, incorporating new data ; however, at the end, the result is not used to provide any real-world application. With regard to the applicants argument that "Applicants fails to recognize how claimed subject matter may be directed to concrete results without being directed to tangible results (page 11, ln. 17-18). The examiner respectfully suggests the applicant to review the Interim Guidelines for examination of Patent Applications for Patent Subject Matter Eligibility, specially pages 19-22.

Regarding claim 25: Claim 25 recites nothing but the physical characteristics of a form of energy and as such are nonstatutory natural phenomena and further more, it

does not appear that a claim reciting a signal encoded with functional descriptive material falls within any of the categories of patentable subject matter set forth in section 101 (Interim Guidelines, Annex IV page 55).

Conclusion

3. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nhon T. Diep whose telephone number is 571-272-7328. The examiner can normally be reached on m-f.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mehrdad Dastouri can be reached on 571-272-7418. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ND
2/20/07

NHON DIEP
PRIMARY EXAMINER

Date Mailed: November 21, 2006

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TAPA.004PA	Application Number: 10/306,942
	Applicant: HANNUKSELA	
	Filing Date: 11/29/02	Group Art Unit: 2621

OIPE NOV 24 2006 IAP43 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
ND	RU 2 123 769	20.12.1998	Russia			Abstract Only	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER /Nhon Diep/	DATE CONSIDERED 02/20/2007

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

TAPA.004

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	KOLS.021PA	3642

7590 09/11/2007

Hollingsworth & Funk, LLC
Suite 125
8009 34th Avenue South
Minneapolis, MN 55425

EXAMINER	
DIEP, NHON THANH	

ART UNIT	PAPER NUMBER
2621	

MAIL DATE	DELIVERY MODE
09/11/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/306,942	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	Nhon T. Diep	2621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 1 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U S C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1 704(b).

Status

1)☒ Responsive to communication(s) filed on *04 June 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24,26 and 27* is/are pending in the application.

4a) Of the above claim(s) *14* is/are withdrawn from consideration.

5)☒ Claim(s) *1-13,15-24,26 and 27* is/are allowed.

6)☐ Claim(s) _____ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☒ Claim(s) *14* are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 November 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Election/Restrictions

1. This application is in condition for allowance except for the presence of claim 14 directed to an invention non-elected with traverse in the reply filed on 5/8/2006. Applicant is given ONE MONTH or THIRTY DAYS from the date of this letter, whichever is longer, to cancel the noted claims or take other appropriate action (37 CFR 1.144). Failure to take action during this period will be treated as authorization to cancel the noted claims by Examiner's Amendment and pass the case to issue. Extensions of time under 37 CFR 1.136(a) will not be permitted since this application will be passed to issue.

The prosecution of this case is closed except for consideration of the above matter.

Allowable Subject Matter

2. Claims 1-13, 15-24 and 26-27 are allowed.

Conclusion

3. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nhon T. Diep whose telephone number is 571-272-7328. The examiner can normally be reached on m-f.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mehrdad Dastouri can be reached on 571-272-7418. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ND
9/3/2007

NHON DIEP
PRIMARY EXAMINER

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	KOLS.021PA	3642

7590 08/10/2006

Hollingsworth & Funk, LLC
Suite 125
8009 34th Avenue South
Minneapolis, MN 55425

EXAMINER	
DIEP, NHON THANH	
ART UNIT	**PAPER NUMBER**
2621	

DATE MAILED: 08/10/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/306,942	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	Nhon T. Diep	2621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 May 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

4a) Of the above claim(s) *14* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) *1-3,15,16,19 and 21-27* is/are rejected.

7)☒ Claim(s) *17,18 and 20* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 November 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *5/06; 3, 5/03; 12/*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Election/Restrictions

1. Applicant's election without traverse of 1-13 and 15-25 in the reply filed on 5/8/2006 is acknowledged. The traversal is on the ground(s) that the examiner fails to identify how the two alleged inventions are disclosed as not capable of use together. This is not found persuasive because as indicated in the election requirement, Claims 1-13, 15-25, drawn to a method of coding video frame for forming a scalable, compressed video sequence, classified in class 375, subclass 240.12 and these claims exclusively claim the steps of forming into the video sequence comprising of a first sub-sequence, a second sub-sequence, the interdependency of the first and the second sub-sequence and forming into the video sequence information indicating which video frames belong to the sub-sequences. Basically, group I has nothing to do with bit rate control but the interdependency of the sub-sequences and group II drawn to a method of transmitting compressed video sequence and to control the transferred bit rate, classified in class 375, subclass 240.16. Therefore, the inventions are distinct, and because these inventions are independent or distinct for the reasons given above and have acquired a separate status in the art in view of their different classification, and the inventions require a different field of search (see MPEP § 808.02) and have acquired a separate status in the art because of their recognized divergent subject matter restriction for examination purposes as indicated is.

The requirement is still deemed proper and is therefore made FINAL.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1-2, 11, 23-24 and 26-27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Regarding claims 1-2, 11, 23 and 26: The method claim of the above claims comprises the steps of forming sub-sequences and the interdependency of these sub sequences without claiming what to do with the formed sub-sequences. The claims, although have useful and concrete results, lack tangible result and therefore these claims do not provide a practical application and are directed to non-statutory subject matter.

Regarding claims 23-24: Claims 23-24 direct to "A computer program for encoding and decoding ..., wherein the computer program comprises program codes...". Since the computer program comprising program codes as claimed does not specify that the computer program is stored (or encoded or embedded) in the computer readable medium and when executed by the computer, the computer program code forming ....The claim computer program of claims 23-24 is therefore a computer per se and is directed to non-statutory subject matter.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application

> by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

5. Claims 1-9, 11-13, 15-16, 19, 22-27 are rejected under 35 U.S.C. 102(e) as being anticipated by Wu et al (US 6,614,936).

Wu et al discloses a system and method for robust video coding using progressive fine-granularity scalable coding comprising the same method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second fame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video fames (col. 7, ln. 63 – col. 8, ln. 5), the method comprising:

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format (figure 11, base layer 332); and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and

at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence (2nd – 3rd frames of the 1st layer 334); and forming into the video sequence information indicating which video frames belong to the second sub-sequence (2nd – 3rd frames belong to the 1st layer and this information is transmitted to the decoding side of the system so decoder can decode) as specified in claims 1, 15, 23 and 25-26; claims 16, 22, 24 are rejected as a reverse process of the encoding process of claim 1); further comprising determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence (arrow between 1st frame of base layer and 2nd frame of 1st layer determines the interdependency between two subsequences) as specified in claim 2; further comprising coding the video sequence into a plural number of scalability layers; and determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from a group comprising: a video frame of a higher scalability layer, a video frame of another sub-sequence in the same scalability layer (fig. 11, base layer to 5th layer and arrow between 1st frame of base layer and 2nd frame of 1st layer determines the interdependency between two subsequences) as specified in claim 3; 4. (Original) A method according to claim 3, further comprising determining the dependencies of the video frames of the second sub-sequence on the basis of at least a scalability layer identifier and a sub-sequence identifier (the dependency of the 3rd frames to the 2nd frame (sub-sequence) of the 1st layer (a scalability layer) as specified in claim 4; further comprising coding the first scalability layer of the video sequence to

comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence (is frame of the base layer) as specified in claim 5; further comprising coding the first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format (1st frame, 2nd and 3rd fames of the base layer) as specified in claim 6; further comprising determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number; or determining a unique identifier for each video frame on the basis of the image number (1st frame of the base layer is known as base layer, 1st sub-sequence and 1st frame) as specified in claims 7-8; further comprising adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence (inherently included so decoder at the decoding side would know how to reconstruct the images) as specified in claim 9; wherein the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image (is frame of the base layer and 2nd frame of the 1 layer) as specified in claim 11; further comprising coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping (1st frame of the base layer to 1st frame of the 1st layer and 1st frame of the base layer to 2nd frame of the 1st layer to 3rd frame of the 1st layer) as specified in claim 12; further comprising coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific (col. 7, ln. 63-64) as specified in claim 13; further comprising

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence or wherein a removal of the second subsequence from the bitstream does not prevent decoding of the bitstream correctly (frames 1-5 of the base layer can be decoded without the need of decoding sub0sequence of frames 2-3 of the 1st layer and therefore the removal of this subsequence does not have any effect in decoding frames 1-5 of the 1st layer) as specified in claims 19 and 27.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 10 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wu et al.

As applied to claim 7 above, it is noted that Wu et al does not particularly disclose the step of adding the identifier to the Supplemental Enhancement Information (SEI) data structure to be transmitted synchronously with the video sequence. Since SEI mechanism is known to proves data delivery that is transferred synchronously with the video data content and therefore, it would have been obvious to one of ordinary skilled in the art at the time the invention was made to modify the system of Wu et al by using SEI mechanism to transfer any identifier data.

With regard to claim 21: Since, the H.26L standard defines a picture order count as a picture position in output order. The decoding process specified in the H.26L

standard uses picture order counts to determine default index orderings for reference pictures in B slices, to represent picture order differences between frames and fields for vector scaling in motion vector prediction and for implicit mode weighted prediction in B slices, and to determine when successive slices in decoding order belong to different pictures. The picture order count is coded and transmitted for each picture. Therefore, it would have been obvious to one of ordinary skilled in the art having both Wu et al and the teaching of the H.26L in front of him/her would have found it obvious to use the picture order count to identify at least partly temporally overlapping sub-sequence frames from the video sequence, and to output from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.

Allowable Subject Matter

8. Claims 17-18 and 20 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The following is a statement of reasons for the indication of allowable subject matter: The various prior art of record does not anticipate or render obviousness steps of entering video frames into a sliding buffer memory in connection with decoding, further comprising decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional, configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers

and continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers as specified in claim 20 and steps of initiating the decoding from a random point in the video sequence; determining an independently decodable sub-sequence that is next after the random point, and setting the value of the image number of the first video frame in the sub-sequence at zero as specified in claim 20.

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Harza (US 6,510,553) discloses a method of streaming video from multiple sources over a network.

b. Radha et al (US 6,639,943) discloses a hydrid temporal-SNR fine granular scalability video coding.

c. Tillman et al (US 6,496,980) discloses a method of providing replay on demend for streaming digital multimedia.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nhon T. Diep whose telephone number is 571-272-7328. The examiner can normally be reached on m-f.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mehrdad Dastouri can be reached on 571-272-7418. The fax phone

number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ND
8/6/2006

NHON DIEP
PRIMARY EXAMINER

Date Mailed: December ____, 2002

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: KOLS.021PA	Application Number:
	Applicant: HANNUKSELA	
	Filing Date: 11/29/02	Group Art Unit: 2621

OIPE JC16 DEC 16 2002 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
ND	6,108,382	8/22/2000	GRINGERI et al.			
ND	5,699,476	12/16/1997	VAN DER MEER			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION YES	TRANSLATION NO
ND	WO 00/67469	9.11.2000	PCT			n/a	
ND	WO 02/054776	11.7.2002	PCT			n/a	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

ND	9-17 Oct. 2002	Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, *Editor's Proposed Draft Text Modification for Joint Video Specification (ITU-T Rec. 264 ISO/IEC 14496-10 AVC)*, Geneva modifications draft 37.)
ND	15 Jan. 1998	ITU-Telecommunications Standardization Union, *Draft Text of Recommendation H.263 Version 2 ("H.263+") for Decision.*
ND	7/10/02	ITU-Telecommunications Standardization Union, *H.26L Test Model Long Term Number 8 (TML-8) Draft0*, Document VCEG-N10. pp. 1-46.
ND	Nov. 2000	ITU-Telecommunications Standardization Union, *Draft for "H.263++" Annexes U,V, and W to Recommendation H.263.*
ND	12 Nov. 2002	David Singer and Toby Walker, *Study Text of ISO/IEC 14496-15/FCD*, pp. 1-25.
ND	12 Nov. 2002	David Singer and Toby Walker, *Study Text of ISO/IEC 14496-15/FCD (Revised)*, pp. 1-26.

EXAMINER /Nhon Diep/	DATE CONSIDERED 06/22/2006

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Date Mailed: May 13, 2003

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: KOLS.021PA	Application Number: 10/306,942
	Applicant: HANNUKSELA	
	Filing Date: 11/29/02	Group Art Unit: ~~2125~~ 2621

OIPE JC42 MAY 27 2003 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
ND	6,337,881	1/8/2002	CHADDHA			
ND	6,307,886	10/23/2001	WESTERMANN			
ND	5,774,593	6/30/1998	ZICK ET AL.			

RECEIVED
MAY 2 9 2003
Technology Center 2100

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION YES	TRANSLATION NO
ND	WO 01/47283	28.06.2001	PCT			N/A	
ND	EP 1 005 232	31.05.2000	EP			N/A	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER /Nhon Diep/	DATE CONSIDERED 06/22/2006

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Date Mailed: March 14, 2003

Sheet 1 of 1

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: KOLS.021PA	Application Number: 10/306,942
	Applicant: HANNUKSELA	
	Filing Date: 11/29/02	Group Art Unit: ~~2135~~ 2621

OIPE
MAR 2 1 2003
PATENT & TRADEMARK OFFICE

RECEIVED
MAR 24 2003
Technology Center 2100

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION YES	TRANSLATION NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

ND	Sept. 2002	Hannuksela, M et al.; Sub-Picture: ROI Coding and Unequal Error Protection, IEEE 2002 Conference, September 2002, USA (internet publication).
ND	Sept. 2002	Hannuksela, M et al.; Sub-Picture: Video-Coding and Unequal Error Protection, XI European Signal Processing Conference, September 2002, Toulouse, France (internet publication).
ND	7.12.2001	Wang, Y.K. et al.: Core Experiment Description of Sub-Picture Coding, ITU 15. Meeting 3.- 7.12.2001, Pattaya, Thailand (internet publication)

EXAMINER /Nhon Diep/	DATE CONSIDERED 06/22/2006

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Date Mailed: May 5, 2006

FORM 1449* IAP21 INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TAPA.004PA	Application Number: 10/306,942
	Applicant: HANNUKSELA	
	Filing Date: 11/29/02	Group Art Unit: ~~2613~~ 2621

OIPE MAY 08 2006 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

ND	2001	B.A. Lokshin, "Digital Broadcasting: from Studio to Audience" Cyrus Systems, Moscow, 2001. (translation included)

EXAMINER /Nhon Diep/	DATE CONSIDERED 06/22/2006

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Notice of References Cited	Application/Control No. 10/306,942	Applicant(s)/Patent Under Reexamination HANNUKSELA, MISKA	
	Examiner Nhon T. Diep	Art Unit 2621	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,614,936	09-2003	Wu et al.	382/238
*	B	US-6,639,943	10-2003	Radha et al.	375/240.11
*	C	US-6,510,553	01-2003	Hazra, Rajeeb	725/87
*	D	US-6,496,980	12-2002	Tillman et al.	725/90
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	TAPA.004PA	3642

76385 7590 04/09/2009

Hollingsworth & Funk, LLC
8009 34th Avenue South
Suite 125
Minneapolis, MN 54425

EXAMINER	
DIEP, NHON THANH	
ART UNIT	PAPER NUMBER
2621	

MAIL DATE	DELIVERY MODE
04/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/306,942	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	Nhon T. Diep	2621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 March 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13,15-24,26 and 27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *15 and 21-24* is/are allowed.

6)☒ Claim(s) *1-13,16-21,26 and 27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *11/29/2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claim 14 is cancelled as per paragraph of the previous Office Action mailed out on 9/11/2007.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1-13, 16-21 and 26-27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Regarding to claims 1-13, 16-21 and 26-27, which are method claims, and that the claims do not fall within one of the four statutory categories of inventions. Supreme Court precedent and recent Federal Circuit decisions indicate a statutory "process" under 35 U.S.C. 101 must (1) be tied to another statutory category (such as a particular apparatus), or (2) transform underlying subject matter (such as an article or material) to a different state or thing. While the instant claim(s) recite a series of steps or acts to be performed, it is considered that none of the steps in the claims positively recites the other statutory class (the thing or product) to which it is tied, for example, by identifying the apparatus that accomplishes the method steps, In re Bilski, 88 USPQ2d 1385 (Fed. Cir. 2008).

Allowable Subject Matter

4. Claims 15, 22-24 are allowed.

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nhon T. Diep whose telephone number is 571-272-7328. The examiner can normally be reached on m-f.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mehrdad Dastouri can be reached on 571-272-7418. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ND

/Nhon T Diep/
Primary Examiner, Art Unit 2621

Date Mailed: October 17, 2007

FORM 1449 E	INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TAPA.004PA	Application Number: 10/306,942
		Applicant: HANNUKSELA	
		Filing Date: Nov. 29, 2002	Group Art Unit: 2621

OIPE
OCT 22 2007
PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/ND/	5,144,426	09.01.1992	Tanaka et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/ND/	01/84850	8 Nov. 2001	WO			NA	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	/Nhon Diep/	DATE CONSIDERED	04/07/2009

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Date Submitted: March 12, 2009 Sheet 1 of 1

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TAPA.004PA	Application Number: 10/306,942
	Applicant: HANNUKSELA	
	Filing Date: 11/29/2002	Group Art Unit: 2621

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/ND/	EP 0798932	10/1/1997	EP			NA	
/ND/	CN 1173783A	2/18/1998	China				X
/ND/							

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	/Nhon Diep/	DATE CONSIDERED	04/07/2009

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Date Mailed: June 10, 2008

Sheet 1 of 1

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TAPA.004PA	Application Number: 10/306,942
	Applicant: HANNUKSELA	
	Filing Date: Nov. 29, 2002	Group Art Unit: 2621

OTPE IAP76 JUN 16 2008 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/ND/	2001/0024472	09/27/2001	Sporer et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER /Nhon Diep/	DATE CONSIDERED 04/07/2009

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

76385 7590 12/07/2009

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
DIEP, NHON THANH	
ART UNIT	PAPER NUMBER
2621	

DATE MAILED: 12/07/2009

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	TAPA.004PA	3642

TITLE OF INVENTION: GROUPING OF IMAGE FRAMES IN VIDEO CODING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	03/08/2010

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

76385 7590 12/07/2009

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	TAPA.004PA	3642

TITLE OF INVENTION: GROUPING OF IMAGE FRAMES IN VIDEO CODING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	03/08/2010

EXAMINER	ART UNIT	CLASS-SUBCLASS
DIEP, NHON THANH	2621	375-240120

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
☐ Issue Fee
☐ Publication Fee (No small entity discount permitted)
☐ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
☐ A check is enclosed.
☐ Payment by credit card. Form PTO-2038 is attached.
☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	TAPA.004PA	3642

76385 7590 12/07/2009

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
DIEP, NHON THANH	
ART UNIT	PAPER NUMBER
2621	

DATE MAILED: 12/07/2009

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 791 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 791 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Notice of Allowability	Application No.	Applicant(s)	
	10/306,942	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	Nhon T. Diep	2621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *7/9/2009*.

2. ☒ The allowed claim(s) is/are *1-13, 15-24, and 26-27, which were renumbered as 1-25, respectively*.

3. ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☒ All b) ☐ Some* c) ☐ None of the:
      1. ☒ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/Nhon T Diep/
Primary Examiner, Art Unit 2621

REASONS FOR ALLOWANCE

1. The following is an examiner's statement of reasons for allowance: The present invention relates to a coding of a video sequence comprising of a plurality of sub-sequence formed therein, at least part of the sub-sequence being formed by coding the intra (I) frames of the at least first frame format, and at least the second frame format, at least part of which is formed by coding video frame of the at least second frame format (P frames), and the identifying of data of the video frames comprised by at least the second sub-sequence are determined into the video sequence in question, and the sub-sequence will comprise information of all sub-sequences that have been directly used for predicting the image frame comprised by the sub-sequence in question, and this information is signaled in the bit stream of the video sequence, separate from the actual image information. The various prior art of record fails to anticipate or render obviousness the encoding of a first sub-sequence information to indicate the interdependency between sub-sequences or to indicate the type of frames, the type of layer, the number of sub-sequence in the layer, the order of frames in the sub-sequence (information of all sub-sequences that have been directly used for predicting the image frame comprised by the sub-sequence in question), and a similar second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from the at least one video frame of the first sub-sequence, and encoding into the video sequence information indicating which video frames belong to the second sub-sequence as specified in claims 1, 15, 16, 22, 23, 24 and 26.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

2. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nhon T. Diep whose telephone number is 571-272-7328. The examiner can normally be reached on m-f.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mehrdad Dastouri can be reached on 571-272-7418. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ND /Nhon T Diep/

Primary Examiner, Art Unit 2621

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TAPA.004PA	Application Number: 10/306,942
	Applicant: HANNUKSELA	
	Filing Date: 11/29/2002	Group Art Unit: 2621

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/ND/	6,266,158	7/24/2001	Hata et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/ND/	10-271511	10/09/1998	JP				X

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	/Nhon Diep/	DATE CONSIDERED	11/21/2009

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Electronic Acknowledgement Receipt

EFS ID:	7035961
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Steven R. Funk
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	18-FEB-2010
Filing Date:	25-JAN-2006
Time Stamp:	19:09:14
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-8-TPO004C1.pdf	54739 5ada0a2bd80887e8733aa6f4751816ecbb49e20a	no	2

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	1449-8-TPO004C1.pdf	62027 dd2dc74c9757a7e106756fb4683ae94b6590b7b1	no	2
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	Elloumi.pdf	337636 7a38047d06aa5d3f7cd8e9e602921ea5c5b034ff	no	4
Warnings:					
Information:					
4	NPL Documents	Hannuksela.pdf	712317 50ee279cd1a1b1c7551108ba7cbed2abfb2c5507	no	14
Warnings:					
Information:					
5	NPL Documents	Soderquist.pdf	641172 08a2996ec505864cfaa3f7b8fa668d9f73f3a2ca	no	6
Warnings:					
Information:					
6	NPL Documents	EPSR-1.pdf	1448766 ac197f5e58085b037546485233f2c1327fd0f075	no	5
Warnings:					
Information:					
7	NPL Documents	EPSR-2.pdf	1732473 a0aa0d91756e7d74226ada4a85633e6f3969b1a9	no	5
Warnings:					
Information:					
8	Foreign Reference	EP0898427.pdf	2658400 a4a37ba6cebd0b966345da06aea44c900df3acab	no	60
Warnings:					
Information:					
9	NPL Documents	TAPA004PA-RR16032006.pdf	589717 dd9806cb4b2d7d65b32b33c965fa551ab50f97cb	no	13
Warnings:					
Information:					
10	NPL Documents	TAPA004PA-OAR-13032007.pdf	543295 a8a2484b1702714bbea0af5d0d0a916eda91a5b7	no	12

Warnings:					
Information:					
11	NPL Documents	TAPA004PA-OAR-09042009.pdf	535372 2f1206b68db40e4ca8daad28ab010b904e168d6a	no	12
Warnings:					
Information:					
12	NPL Documents	TAPA004PA-OAR10082006.pdf	808657 c424432fc9383467ba6351f18e30eff8eac8e64d	no	16
Warnings:					
Information:					
13	NPL Documents	TAPA004PA-OA16032006.pdf	169633 a6390f80af78ba4d9b79d8573a0af449252ae4bb	no	4
Warnings:					
Information:					
14	NPL Documents	TAPA004PA-OA13032007.pdf	240311 b4f05eecd63d96dcc100f84be33ae6d606179a02	no	5
Warnings:					
Information:					
15	NPL Documents	TAPA004PA-OA11092007.pdf	160508 fe1556ebcf37a1e60a1f35d6e46d29b7f2fe0fcd	no	4
Warnings:					
Information:					
16	NPL Documents	TAPA004PA-OA10082006.pdf	763601 b767b90add0d470accc93e5830f1b4af899fd807	no	16
Warnings:					
Information:					
17	NPL Documents	TAPA004PA-OA09042009.pdf	328025 ee2eb11396b0d32275b729a41dcab7d02c43f190	no	7
Warnings:					
Information:					
18	NPL Documents	TAPA004PA-Allowance.pdf	507388 8687fa54e8edcd1b0f4307d369d6574904ccf80e	no	8
Warnings:					
Information:					
		Total Files Size (in bytes):	12294037		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,996	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of the foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided. Also attached is a copy of the English translation of the Japanese Office Action.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to

establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Steven R. Funk/
Steven R. Funk
Reg. No. 37,830

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5122875	06/16/1992	Raychaudhuri et al.			
	5838265	11/17/1998	Adolph			
	5877812	03/02/1999	Krause et al.			
	6314139	11/06/2001	Koto et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	02/04/2010	Copy of Allowance Notification with translation dated 2/4/2010 from parallel Russian Application No. 2006110322, 17 pages.
	02/25/2010	Copy of translated Office Action dated 2/25/2010 from parallel Japanese Application No. 2006-120297, 5 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Electronic Acknowledgement Receipt

EFS ID:	7316482
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Steven R. Funk
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	30-MAR-2010
Filing Date:	25-JAN-2006
Time Stamp:	15:42:54
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-9-TPO004C1.pdf	54556 ba342520fa59d88ef6da05e0787eea587f2da67e	no	2

Warnings:	
Information:	

2	NPL Documents	OA20100225.pdf	98113 56c0b5dbea6eaa02a0dd55397547b2bad6afd76f	no	5
Warnings:					
Information:					
3	NPL Documents	RUAllowance.pdf	2428420 81a2a0594f6e1200fcb4ddfe9abda3436c7a50f9	no	17
Warnings:					
Information:					
4	Information Disclosure Statement (IDS) Filed (SB/08)	1449-9-TPO004C1.pdf	52394 0584687ae502ebf310fc97fc2383dfae0dcd98e3	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
			Total Files Size (in bytes):	2633483	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,996	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of the foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a

reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Steven R. Funk/
Steven R. Funk
Reg. No. 37,830

Reference Number: B065214 Dispatch Number: 141304

Mailing Date: March 2, 2010

NOTICE OF REASONS FOR REJECTION

(TRANSLATION)

Patent Application No.:	2006-120297
Drafting Date:	February, 2010
Patent Office Examiner:	Toshio Sakamoto 2954 5C00
Attorney for Patent Applicant:	Atsushi Aoki (and four others)
Applicable Provisions:	Article 29, paragraph 1; Article 29, paragraph 2; Article 36

It is deemed that the present application should be rejected for the following reasons. An argument, if any, should be submitted within 3 months from the mailing date of this notice.

Reasons

a. The inventions described in the following claims of this application are inventions described in the following publications distributed or the inventions made available to the public through electronic telecommunications lines in Japan or a foreign country prior to the filing of the patent application and, therefore, are unpatentable under the provisions of Article 29, paragraph 1, item 3 of the Patent Law.

NOTE (For the numbers of the cited references, see the List of Cited References)

Claims: 1, 5, 9 to 10, and 13 to 14
Reference: 1

Remarks:

It is deemed that there is no difference between the inventions according to claims 1, 5, 9 to 10, 13, and 14 of the present application and the invention described in Reference 1 (refer to Reason b. below).

b. The inventions described in the following claims of this application are deemed ones which could easily have been made, prior to the filing of the patent application, by a person with ordinary skill in the art to which the inventions pertain, on the basis of the inventions described in the following publications distributed or the inventions made available to the public through electronic telecommunications lines in Japan or a foreign country prior to the filing of the patent application and, therefore, are unpatentable under the provisions of Article 29, paragraph 2, of the Patent Law.

NOTE (For the numbers of the cited references, see the List of Cited References)

Claims: 1 to 14
Reference: 1

Remarks:

Reference 1 describes a coder apparatus for interframe

predictive coding, wherein information of an editable point, which is the first coded picture of a closed GOPs, is multiplexed in a header area (specifically refer to paragraphs [0001] to [0003], [0032], and [0064]).

c. This application does not satisfy the requirements prescribed in Article 36, Paragraph 6, Item 2 of the Patent Law on the following point.

NOTE

(1) Claim 1 describes "at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order", however, it is unclear as to what image frame is referred by the "previous" image frame, therefore, the referential relationship as well as the temporal relationship among "at least one reference image frame", "the previous reference image frame in decoding order", and "at least one previous image frame" are unclear.

Specifically, it is unclear as to whether the above description means that one reference image frame is predictable from an image frame that is earlier in decoding order than said reference image frame, or that one reference image frame is predictable from an earlier image frame in decoding order, and said earlier image in decoding order is also predictable from an earlier image frame in decoding order.

The descriptions in claims 5, 9 to 10, 13, and 14 are likewise unclear.

(2) Claim 1 describes "the first image frame is an image frame of the independently decodable sequence of image frames which is to be decoded first in decoding order", however, it is unclear as to in what "first" is, for example, whether "first" means "first" in "the independently decodable sequence" or not, etc.

The descriptions in claims 3, 5, 7, 9, 10, and 12 to 14 are likewise unclear.

(3) Claim 9 describes "A computer program comprising computer program code for use with a computer, the computer program code comprising ... (snip) ..., wherein ... (snip) ..., characterized in that the computer program comprises code for ... (snip) ..., and wherein ... (snip) ...,being included in the sequence." However, it is unclear as to (i) how the terms "program" and "code" are distinguished from one another, and (ii) whether or not "program" is used to cause a computer to execute a process (refer to "Examination Guidelines for Patent and Utility Model" PART VII, Chapter 1, 1.1.1(2)(b)).

The same is true for claim 14.

Note that, if the above description of claim 9 is amended to read "A program in accordance with which a computer executes a process of coding ... (snip) ..., wherein the process comprises a step for V-ing ... (snip) ...", the above objection will overcome.

The same is true for claim 14.

If an amendment is made, it should be noted that no new matter can be added, and in an argument, the reasons that the amendments meet the requirements of the Patent Law should be

stated, identifying the relevant descriptions of the original specification on which the amendments are based.

If another reason for rejection is found, a further notice of reasons for rejection will be issued.

List of Cited References

1. Japanese Unexamined Patent Publication (Kokai) . 11-88874

Record of Results of Prior Art Search

· Searched field IPC

H04N 7/24-7/68

This Record of Results of Prior Art Search does not constitute a reason for rejection.

For inquiries on the content of this Notice of Reasons for Rejection or a request for an interview on this case, please contact the Examiner below.

Electronic Image Devices Division,
Fourth Patent Examination Department
Toshio Sakamoto
TEL. 03 (3581) 1101 extension 3539
FAX. 03 (3501) 0715

TRANSLATION

FEDERAL SERVICE ON INDUSTRIAL PROPERTY, PATENTS AND TRADE MARKS (ROSPATENT)

To No. 2412-136662RU/8132
Our No. 2006110322/09(011235)

DECISION ON GRANT OF A PATENT FOR INVENTION

(21) Application No.: 2006110322/09(011235) (22) Application filing date: 30 March 2006

Substantive examination of the present application for invention has stated that

[ ] the claimed invention

[x] the claimed group of inventions

relates to the objects of intellectual rights and complies with the patentability conditions stipulated by the Civil Code of the Russian Federation and, in view of this, a decision to grant a patent for the invention has been taken.

A Report on Examination Results is enclosed.

Head *Signature* S.P. Simonov

Report on Examination Results

(21) Application № 2006110322/09(011235). (22) Date of filing the application 30 March 2006
(24) Date from which industrial property rights may have effect 22 January 2003
(85) Date of commencement of the national phase

PRIORITY IS FIXED ON DATE

☐ (22) Date of filing the application
☐ (23) Date of filing of additional materials of to the earlier application №
☐ (62) ☐ priority date of the application № 2004125579 of 22 January 2003 from which the present application has been divided up
☐ filing date of the application № of from which the present application has been divided up
☐ (66) Filing date of the earlier application №
☐ (30) Data relating to priority under the Paris Convention

(31) Number assigned to priority application	(32) Date of filing priority application	(33) Country code	Claim
20020127	21 January 2002	FI	

(86) PCT Application number and date of .
(87) PCT Publication number and date of .

(72) Inventor(s) HANNUKSELA, Miska, FI
(73) Assignee(s) NOKIA CORPORATION, FI
(54) Title GROUPING OF IMAGE FRAMES IN VIDEO CODING

The Examination department basing on the results of substantive examination of the patent application conducted in respect to

☐ originally filed claims ☒ claims amended by the applicant

has revealed their concordance to the requirements of patentability set forth by the Articles 1349 and 1350 of the Civil Code of the Russian Federation and decided to grant the Patent of the Russian Federation for the following claims:

(21) 2006110322/09

(51) IPC

H04N 7/26 (2006.01)

H04L 13/08 (2006.01)

(57)

1. A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, characterized by

decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames;

configuring in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames; and

using the image frames in the buffer memory in the decoding process.

2. A method according to claim. 1, characterized in that said indication informs that at least one discontinuity in the numbering of the image frames in the video sequence are intentional.

3. A method according to claim 1, characterized by further comprising

using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

4. A method according to claim 1, characterized in that the number of image frames in the buffer memory are used in sliding window buffering of image frames.

5. A method according to claim 1 characterized by further comprising

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames.

6. A method according to claim 1, characterized by further comprising

entering into the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames.

7. A method according to claim 6 characterized in that said filler frames are designated by an identifier indicating that the filler frames do not belong to the actual video sequence.

8. A method according to claim 1, characterized in that the step of configuring the buffer memory to provide a number of image frames corresponding to the discontinuities in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.

9. A. method according to claim 3 or 6, characterized by further comprising

detecting an error in the compressed video sequence on the basis of a reference in the decoding process to an image frame corresponding to a discontinuity in the numbering of image frames.

10. A video decoder for decoding a compressed video sequence,

the video decoder being, connected to a buffer memory wherein image frames are entered in connection with the decoding, characterized in that the video decoder is arranged to

decode from the video sequence an indication related to at least one discontinuity of numbering of the image frames;

configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames; and

use the image frames in the buffer memory in the decoding process.

11. A video decoder according to claim 10, characterized in that said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional.

12. A video decoder according, to claim 10, characterized in that the video decoder is further arranged to

use the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

13. A video decoder according to claim 10 characterized in that the number of image frames in the buffer memory are used in sliding window buffering of image frames.

14. A video decoder according to claim 10, characterized in that the video decoder is further arranged to

continue the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames.

15. A video decoder according to claim 10, characterized in that the video decoder is further arranged to

enter into the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames.

16. A video decoder according to claim 15, characterized in that

said filler frames are designated by an identifier indicating that the filler frames do not belong to the actual video sequence.

17. A video decoder according to any of the claims 10, characterized in that the video decoder is further arranged to

configure the buffer memory to provide a number of image frames corresponding to the discontinuities in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

18. A computer readable medium having computer program code stored thereon and executable in a data processing device, the computer program code for decoding a compressed video sequence, image frames of said video sequence being entered into a buffer memory in connection with the decoding, characterized in that the computer program code comprises

a computer program code for decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames;

a computer program code for configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames; and

a computer program code for including the image frames in the buffer memory in the decoding process.

19. A computer readable medium according to claim 18, characterized in that said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional.

20. A computer readable medium according to claim 18, characterized in that the computer program product further comprises

a computer program code for including the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

21. A computer readable medium according to claim 18, characterized in that the computer program product further comprises

a computer program code for using the number of image frames in the buffer memory in sliding window buffering of image frames.

22. A computer readable medium according to claim 18, characterized in that the computer readable medium, further comprises

a computer program code for continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames.

23. A computer readable medium according to claim 18, characterized in that the computer readable medium further comprises

a computer program code for entering into the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames.

24. A computer readable medium according to claim 23 characterized in that the computer readable medium further comprises

a computer program code for designating said filler frames by an identifier indicating that the filler frames do not belong to the actual video sequence.

26. A computer readable medium according to claim 18, characterized in that the computer readable medium further comprises a computer program code for configuring the buffer memory to provide a number of image frames corresponding to the discontinuities in the numbering of the image frames by memory indications, whereby no data is enterer, into the buffer memory.

28. A video encoder for encoding a video sequence, characterized in that the video encoder is arranged to

encode the video sequence such that at least one image frame is excluded from the video sequence, causing a discontinuity in the numbering of image frames, and

encode into the video sequence an indication related to the discontinuity of numbering of the image frames.

27. A video encoder according to claim 26, characterized in that said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional.

28. A device for adjusting bit rate of a video sequence, the device being arranged to receive an encoded video sequence, characterized in that the device is further arranged to

exclude at least one image frame from the video sequence, thereby causing a discontinuity in the numbering of image frames, and

include into the video sequence an indication related to the discontinuity of numbering of the image frames.

29. A device according to claim 28, characterized in that said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional.

30. A video signal comprising a compressed video sequence comprising image frames, characterized in that the video signal comprises

an indication related to the discontinuity of numbering of the image frames in the transmitted video signal,

wherein the indication informs a decoder to use in a decoding process a number of image frames corresponding to the discontinuity in the numbering of the image frames.

31. A video signal according to claim 30, characterized in that said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional.

(56)

STEPHAN WENGER, Temporal Scalability using P-Pictures for Low-Latency Applications, IEEE Signal Processing Society 1998 Workshop on Multimedia Signal Processing, Los Angeles, December 7-9, 1998

US 5838265 A, 1998.11.17

US 5877812 A, 1999.03.02

WO 0147283 A1, 2001.06.28

RU 2128405 C1, 1999.03.27

For publication of information on issuance of the patent the specification and the drawings as originally filed shall be used.

Enclosure: The abstract amended by the Examiner on 1 page.

State Patent Examiner

To application No.2006110322/09

(54) GROUPING OF IMAGE FRAMES IN VIDEO CODING

Abstract

(57) The invention relates to the grouping multimedia files, in particular, video files for using in streaming. The technical result is to provide possibility for decoder to take into account frames which a coder has deleted. The indicated technical result is achieved by that the method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding. The video sequence includes an indication related to at least one discontinuity of numbering of the image frames; the indication is decoded from the video sequence. Further in response to the indication, the buffer memory is configured to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames, and the image frames are used in the buffer memory in the decoding process. Preferably the indication informs that at least one discontinuity in the numbering of the image frames in the video sequence is intentional, and the number of image frames is used in the buffer memory in place of image frames that do not exist at a decoder.

6 independent claims, 25 dependant claims, 14 figures.

ФЕДЕРАЛЬНАЯ СЛУЖБА ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ
(РОСПАТЕНТ)
Бережковская наб., 30, корп. 1, Москва, Г-59, ГСП-5, 123995. Телефон (8-499) 240- 60- 15. Факс (8-495) 234- 30- 38

На № 2412-136662RU/8132 от 10.08.2009

Наш № 200611032 2/09(011235)

При переписке просим ссылаться на номер заявки и сообщать дату получения настоящей корреспонденции от

129090, Москва, ул. Б.Спасская, 25, стр.3,
ООО "Юридическая фирма Городисский и Партнеры"

пат.пов. Ю.Д.Кузнецов, рег.N 595

04 ФЕВ 2010

Р Е Ш Е Н И Е
о выдаче патента на изобретение

(21) Заявка № 2006110322/09(011235) (22) Дата подачи заявки 30.03.2006

В результате экспертизы заявки на изобретение по существу установлено, что

[ ] заявленное изобретение

[x] заявленная группа изобретений

относится к объектам патентных прав и соответствует условиям патентоспособности, предусмотренным Гражданским кодексом Российской Федерации, в связи с чем принято решение о выдаче патента на изобретение.

Заключение по результатам экспертизы прилагается.

Приложение: на 8 л. в 1 экз.

Руководитель Б.П.Симонов

Date G&P: 09/02/2010

0005839755

Адрес для переписки с патентообладателем или его представителем, который будет опубликован в официальном бюллетене
[ ]

[x] указан на лицевой стороне решения

Адрес для направления патента
[ ]

[x] указан на лицевой стороне решения
[ ] указан в графе «Адрес для переписки с патентообладателем…»

В результате экспертизы заявки по существу, проведенной в отношении уточненной заявителем формулы изобретения установлено соответствие заявленной группы изобретений требованиям статей 1349 и 1350 Гражданского кодекса Российской Федерации.

Формула изобретения приведена на странице(ах) 3 - 8.

ЗАКЛЮЧЕНИЕ ПО РЕЗУЛЬТАТАМ ЭКСПЕРТИЗЫ

(21) Заявка № 2006110322/09(011235) (22) Дата подачи заявки 30.03.2006

(24) Дата начала отсчета срока действия патента 22.01.2003

ПРИОРИТЕТ УСТАНОВЛЕН ПО ДАТЕ

[x] (62) [x] приоритета изобретения по первоначальной заявке № 2004125579 от 22.01.2003
из которой данная заявка выделена

[x] (30) подачи первой(ых) заявки(ок) в государстве-участнике Парижской конвенции

	(31) Номер первой(ых) заявки(ок)	(32) Дата подачи первой(ых) заявки(ок)	(33) Код страны
1.	20020127	21.01.2002	FI

(72) Автор(ы) ХАННУКСЕЛА Миска, FI

(73) Патентообладатель(и) НОКИА КОРПОРЕЙШН, FI

(54) Название изобретения ГРУППИРОВАНИЕ КАДРОВ ИЗОБРАЖЕНИЯ НА ВИДЕО КОДИРОВАНИИ

(см. на обороте)

06 2	ДОМ 10.08.2009	09.2501

ВНИМАНИЕ! *С целью исключения ошибок просьба проверить сведения, приведенные в заключении, т.к. они без изменения будут внесены в Государственный реестр изобретений Российской Федерации, и незамедлительно сообщить об обнаруженных ошибках.*

Форма № 01а

(21) 2006110322/09

(51) МПК
H04N 7/26 (2006.01)
H04L 13/08 (2006.01)

(57)

1. Способ декодирования сжатой видеопоследовательности, при этом кадры изображения вводятся в буферную память в связи с декодированием, отличающийся тем, что содержит этапы, на которых декодируют из видеопоследовательности индикацию, относящуюся к по меньшей мере одному разрыву в нумерации кадров изображения; конфигурируют, в ответ на эту индикацию, буферную память так, чтобы она обеспечивала количество кадров изображения, соответствующее разрыву в нумерации кадров изображения; и используют кадры изображения в буферной памяти в процессе декодирования.

2. Способ по п.1, отличающийся тем, что упомянутая индикация информирует о том, что по меньшей мере один разрыв в нумерации кадров изображения в видеопоследовательности является преднамеренным.

3. Способ по п. 1, отличающийся тем, что дополнительно содержит этап, на котором используют упомянутое количество кадров изображения в буферной памяти вместо кадров изображения, которых нет на декодере вследствие разрывов в нумерации кадров изображения.

4. Способ по п. 1, отличающийся тем, что упомянутое количество кадров изображения в буферной памяти используется при буферизации кадров изображения по принципу скользящего окна.

5. Способ по п. 1, отличающийся тем, что дополнительно содержит этап, на котором продолжают ввод в буферную память кадров изображения, содержащихся в рассматриваемой видеопоследовательности, начиная с правильного номера кадра изображения, после того как буферная память сконфигурирована так, чтобы она обеспечивала количество кадров изображения, соответствующее разрывам в нумерации кадров изображения.

6. Способ по п. 1, отличающийся тем, что дополнительно содержит этап, на котором вводят в буферную память количество заполняющих кадров, соответствующее разрывам в нумерации кадров изображения.

7. Способ по п. 6, отличающийся тем, что упомянутые заполняющие кадры обозначены идентификатором, указывающим, что заполняющие кадры не принадлежат фактической видеопоследовательности.

8. Способ по п. 1, отличающийся тем, что этап конфигурирования буферной памяти так, чтобы она обеспечивала количество кадров изображения, соответствующее разрывам в нумерации кадров изображения, выполняют посредством индикаторов памяти, при этом никакие данные в буферную память не вводятся.

9. Способ по п. 3 или 6, отличающийся тем, что дополнительно содержит этап, на котором обнаруживают ошибку в сжатой видеопоследовательности на основе ссылки в процессе декодирования на кадр изображения, соответствующий разрыву в нумерации кадров изображения.

10. Видеодекодер для декодирования сжатой видеопоследовательности, при этом видеодекодер соединен с буферной памятью, в которую вводятся кадры изображения в связи с декодированием, отличающийся тем, что выполнен с возможностью декодирования из видеопоследовательности индикации, относящейся к по меньшей мере одному разрыву в нумерации кадров изображения; конфигурирования, в ответ на эту индикацию, буферной памяти так, чтобы она обеспечивала количество кадров изображения, соответствующее разрыву в нумерации кадров изображения; и использования кадров изображения в буферной памяти в процессе декодирования.

11. Видеодекодер по п. 10, отличающийся тем, что упомянутая индикация информирует о том, что разрывы в нумерации кадров изображения в видеопоследовательности являются преднамеренным.

12. Видеодекодер по п. 10, отличающийся тем, что дополнительно выполнен с возможностью использования упомянутого количества кадров

изображения в буферной памяти вместо кадров изображения, которых нет на декодере вследствие разрывов в нумерации кадров изображения.

13. Видеодекодер по п. 10, отличающийся тем, что упомянутое количество кадров изображения в буферной памяти используется при буферизации кадров изображения по принципу скользящего окна.

14. Видеодекодер по п. 10, отличающийся тем, что дополнительно выполнен с возможностью продолжать ввод в буферную память кадров изображения, содержащихся в рассматриваемой видеопоследовательности, начиная с правильного номера кадра изображения, после того как буферная память сконфигурирована так, чтобы она обеспечивала количество кадров изображения, соответствующее разрывам в нумерации кадров изображения.

15. Видеодекодер по п.10, отличающийся тем, что дополнительно выполнен с возможностью ввода в буферную память количества заполняющих кадров, соответствующего разрывам в нумерации кадров изображения.

16. Видеодекодер по п.15, отличающийся тем, что упомянутые заполняющие кадры обозначены идентификатором, указывающим, что заполняющие кадры не принадлежат фактической видеопоследовательности.

17. Видеодекодер по п.10, отличающийся тем, что дополнительно выполнен с возможностью конфигурирования буферной памяти так, чтобы она обеспечивала количество кадров изображения, соответствующее разрывам в нумерации кадров изображения, посредством индикаторов памяти, при этом никакие данные в буферную память не вводятся.

18. Машиночитаемый носитель, на котором хранится исполняемый в устройстве обработки данных компьютерный программный код, предназначенный для декодирования сжатой видеопоследовательности, при этом кадры изображения упомянутой видеопоследовательности вводятся буферную память в связи с декодированием, отличающийся тем, что компьютерный программный код содержит компьютерный программный код для декодирования из видеопоследовательности индикации, относящейся к по меньшей мере одному разрыву в нумерации кадров изображения; компьютерный

программный код для конфигурирования, в ответ на эту индикацию, буферной памяти так, чтобы она обеспечивала количество кадров изображения, соответствующее разрыву в нумерации кадров изображения; и компьютерный программный код для включения кадров изображения в буферную память в процессе декодирования.

19. Машиночитаемый носитель по п.18, отличающийся тем, что упомянутая индикация информирует о том, что разрывы в нумерации кадров изображения в видеопоследовательности являются преднамеренным.

20. Машиночитаемый носитель по п.18, отличающийся тем, что дополнительно содержит компьютерный программный код для включения упомянутого количества кадров изображения в буферную память вместо кадров изображения, которых нет на декодере вследствие разрывов в нумерации кадров изображения.

21. Машиночитаемый носитель по п. 18, отличающийся тем, что дополнительно содержит компьютерный программный код для использования упомянутого количества кадров изображения в буферной памяти при буферизации кадров изображения по принципу скользящего окна.

22. Машиночитаемый носитель по п.18, отличающийся тем, что дополнительно содержит компьютерный программный код для продолжения ввода в буферную память кадров изображения, содержащихся в рассматриваемой видеопоследовательности, начиная с правильного номера кадра изображения, после того как буферная память сконфигурирована так, чтобы она обеспечивала количество кадров изображения, соответствующее разрывам в нумерации кадров изображения.

23. Машиночитаемый носитель по п. 18, отличающийся тем, что дополнительно содержит компьютерный программный код для ввода в буферную память количества заполняющих кадров, соответствующего разрывам в нумерации кадров изображения.

24. Машиночитаемый носитель по п.23, отличающийся тем, что дополнительно содержит компьютерный программный код для обозначения

упомянутых заполняющих кадров идентификатором, указывающим, что заполняющие кадры не принадлежат фактической видеопоследовательности.

25. Машиночитаемый носитель по п.18, отличающийся тем, что дополнительно содержит компьютерный программный код для конфигурирования буферной памяти так, чтобы она обеспечивала количество кадров изображения, соответствующее разрывам в нумерации кадров изображения, посредством индикаторов памяти, при этом никакие данные в буферную память не вводятся.

26. Видеокодер для кодирования видеопоследовательности, отличающийся тем, что выполнен с возможностью кодирования видеопоследовательности так, что по меньшей мере один кадр изображения исключен из видеопоследовательности, что приводит к разрыву в нумерации кадров изображения, и кодирования в видеопоследовательность индикации, относящейся к упомянутому разрыву в нумерации кадров изображения.

27. Видеокодер по п. 2 6, отличающийся тем, что упомянутая индикация информирует о том, что разрывы в нумерации кадров изображения в видеопоследовательности являются преднамеренным.

28. Устройство для регулирования скорости передачи в битах видеопоследовательности, выполненное с возможностью приема закодированной видеопоследовательности, отличающееся тем, что дополнительно выполнено с возможностью исключения по меньшей мере одного кадра изображения из видеопоследовательности, тем самым обуславливая разрыв в нумерации кадров изображения, включения в видеопоследовательность индикации, относящейся к упомянутому разрыву в нумерации кадров изображения.

29. Устройство по п. 28, отличающееся тем, что упомянутая индикация информирует о том, что разрывы в нумерации кадров изображения в видеопоследовательности являются преднамеренным.

30. Видеосигнал, содержащий сжатую видеопоследовательность, содержащую кадры изображения, отличающийся тем, что содержит индикацию, относящуюся к разрыву в нумерации кадров изображения в передаваемом

видеосигнале, при этом данная индикация информирует декодер о необходимости использования количества кадров изображения, соответствующего упомянутому разрыву в нумерации кадров изображения, в процессе декодирования.

31. Видеосигнал по п. 30, отличающийся тем, что упомянутая индикация информирует о том, что разрывы в нумерации кадров изображения в видеопоследовательности являются преднамеренным.

(56)
STEPHAN WENGER, Temporal Scalability using P-Pictures for Low-Latency Applications, IEEE Signal Processing Society 1998 Workshop on Multimedia Signal Processing, Los Angeles, December 7 - 9, 1998
US 5838265 A, 1998.11.17
US 5877812 A, 1999.03.02
WO 0147283 A1, 2001.06.28
RU 2128405 C1, 1999.03.27

При публикации сведений о выдаче патента будут использованы первоначальные чертежи и описание.

Приложение: 1.Реферат, скорректированный экспертизой, на 1 л. в 1 экз.

Заместитель заведующего
отделом электрорадиотехники ФГУ ФИПС С.В. Махотина

В.Н.Воропай 8 (495) 730 76 49

К заявке № 2006110322/09

(54) ГРУППИРОВАНИЕ КАДРОВ ИЗОБРАЖЕНИЯ НА ВИДЕО КОДИРОВАНИИ

Реферат

(57) Изобретение относится к группированию мультимедийных файлов, в частности, видео файлов для использования в потоковой передачи. Техническим результатом является обеспечение декодеру возможности учитывать кадры изображения, которые кодер намеренно удалил. Указанный технический результат достигается тем, что способ декодирования сжатой видеопоследовательности, при этом кадры изображения вводятся в буферную память в связи с декодированием. Видеопоследовательность включает в себя индикацию, относящуюся к по меньшей мере одному разрыву в нумерации кадров изображения, причем данная индикация декодируется из видеопоследовательности. Далее, в ответ на эту индикацию буферная память конфигурируется так, чтобы она обеспечивала количество кадров изображения, соответствующее разрыву в нумерации кадров изображения, и кадры изображения в буферной памяти используются в процессе декодирования. Предпочтительно, упомянутая индикация информирует о том, что по меньшей мере один разрыв в нумерации кадров изображения в видеопоследовательности является преднамеренным, и упомянутое количество кадров изображения используется в буферной памяти вместо кадров изображения, которых нет на декодере. 6 н.п., 25 з.п. ф-лы, 14 ил.

Референт В.Н. Воропай

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,996	01/25/2006	Miska Hannuksela	TPO.004.C1	5361

76385 7590 06/17/2010
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

MAIL DATE	DELIVERY MODE
06/17/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/338,996	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	SIU M. LEE	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 January 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6 and 10-31* is/are rejected.

7)☒ Claim(s) *7-9* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 January 2006* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☒ Certified copies of the priority documents have been received in Application No. 10/306942.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *6/19/2006, 8/18/2006, 11/24/2006, 10/22/2007, 6/16/2008, 3/11/2009, 4/20/2009, 2/8/2010, 2/18/2010, 3/30/2010*
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Objections

1. Claims 1, 5, 9, 10, 14, 18, 22, 24, 26, 28, and 30 are objected to because of the following informalities:

(1) Regarding claim 1:

Line 4 recites “decoding from the video sequence an indication”; the examiner assumes "the video sequence" is referring to the compressed video sequence in line 1. The examiner suggests changing "the video sequence" in line 4 to "the **compressed** video sequence” for clarification.

Lines 7-8 recites “corresponding to a discontinuity in the numbering of the image frames”; the examiner assumes “a discontinuity in the numbering” is refer to the “at least one discontinuity of numbering” referred in lines 4-5. The examiner suggests changing “a discontinuity” in line 7 to “the discontinuity”.

(2) Regarding claim 5:

Lines 2-3 recite “the video sequence in question”; there is a lack of antecedent basis for this imitation.

Line 3 recites “the correct image frame number”; there is a lack of antecedent basis for this limitation.

(3) Regarding claim 9:

Line 2 recites “the basis” there is a lack of antecedent basis for this limitation.

(4) Regarding claim 10:

Line 5 recites “decoding from the video sequence an indication”; the examiner assumes "the video sequence" is referring to the compressed video sequence in line 1. The examiner suggests changing "the video sequence" in line 4 to "the **compressed** video sequence” for clarification.

Lines 8-9 recites “corresponding to a discontinuity in the numbering of the image frames”; the examiner assumes “a discontinuity in the numbering” is refer to the “at least one discontinuity of numbering” referred in lines 4-5. The examiner suggests changing “a discontinuity” in line 7 to “the discontinuity”.

(5) Regarding claim 14:

Lines 3-4 recite “the video sequence in question”; there is a lack of antecedent basis for this imitation.

Line 4-5 recite “the correct image frame number”; there is a lack of antecedent basis for this limitation.

(6) Regarding claim 18:

Line 11 recites “corresponding to a discontinuity in the numbering of the image frames”; the examiner assumes “a discontinuity in the numbering” is refer to the “at least one discontinuity of numbering” referred in lines 7. The examiner suggests changing “a discontinuity” in line 7 to “the discontinuity”.

(7) Regarding claim 22:

Line 4 recites “the video sequence in question”; there is a lack of antecedent basis for this limitation.

Line 5 recites “the correct image frame number”; there is a lack of antecedent basis for this limitation.

(8) Regarding claim 24:

Lines 4-5 recite “the actual video sequence”; there is a lack of antecedent basis for this limitation.

(9) Regarding claim 26:

Lines 4-5 recite "the numbering of image frames"; there is a lack of antecedent basis for this limitation.

(10) Regarding claim 28

Line 5 recites "the numbering of image frames"; there is a lack of antecedent basis for this limitation.

(11) Regarding claim 30:

Line 3 recites “the discontinuity of numbering of the image”; there is a lack of antecedent basis for this limitation.

Line 4 recites “the transmitted video signal”; there is a lack of antecedent basis for this limitation.

Appropriate correction is required.

Drawings

2. The drawings are objected to because Figure 1 does not have sufficiently descriptive labels. Blank boxes in drawings should be labeled descriptively unless it is a well-known component. For example, the blank box, "100" should be labeled as

"Multimedia Source". Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 10-17 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claimed "video decoder" is non-structural per se and because the specification discloses a "software" embodiment (page 6,

paragraph 0015 and page 33, paragraph 0097). Therefore, a reasonable interpretation in light of the specification leads to the conclusion that the claim video decoder as pure software, which does not fall within the definition of a process, machine, manufacture or composition of matter.

Claims 18-25 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claim is drawn to a "computer program product stored in a computer readable medium". The specification does not define the computer readable medium, therefore, it includes transitory and non-transitory computer readable medium. The broadest reasonable interpretation of the claim in light of the specification concludes that the claim as a whole covers a transitory signal, which does not fall within the definition of a process, machine, manufacture, or composition of matter (In re Nuijten). Therefore, claim 18-25 do not fall within a statutory category (the rejection can be overcome by specifying the computer readable medium as "non-transitory computer readable medium").

Claims 26-27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claimed "video encoder" is non-structural per se and because the specification discloses a "software" embodiment (page 6, paragraph 0015 and page 33, paragraph 0097). Therefore, a reasonable interpretation in light of the specification leads to the conclusion that the claim video decoder as pure

software, which does not fall within the definition of a process, machine, manufacture or composition of matter.

Claims 28-29 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claimed "device" is non-structural per se and because the specification discloses a "software" embodiment (page 6, paragraph 0015 and page 33, paragraph 0097). Therefore, a reasonable interpretation in light of the specification leads to the conclusion that the claim video decoder as pure software, which does not fall within the definition of a process, machine, manufacture or composition of matter.

Claims 30-31 are directed to a "video signal"; a signal is not limited to that which falls within a statutory category of invention because it is not limit to a process, machine, manufacture, or a composition of matter. Instead, it includes a form of energy. Energy does not fall with in a statutory category since it is clearly not a series of steps or acts to constitute a process, not a mechanical device or combination of mechanical device to constitute a machine, not a tangible physical article or object which is some form of matter to be a product and constitute a manufacture, and not a composite of two or more substances to constitute a composite of matter.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the

art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 26-27 are rejected under 112, 1st paragraph as being a single component (video encoder) claim, i.e., where a single component (video encoder) recitation does not appear in combination with another recited component, is subject to an undue breadth rejection. The single component which covered every conceivable ways for achieving the stated purpose was held nonenabling for the scope of claim because the specification disclosed at most only those known to the inventor. See MPEP §2164.08(a).

Claims 28-29 are rejected under 112, 1st paragraph as being a single component (device) claim, i.e., where a single component (device) recitation does not appear in combination with another recited component, is subject to an undue breadth rejection. The single component which covered every conceivable ways for achieving the stated purpose was held nonenabling for the scope of claim because the specification disclosed at most only those known to the inventor. See MPEP §2164.08(a).

Double Patenting

5. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct

from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

6. Claims 1-5 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 17 of copending Application No. 10/306942.

This is a provisional obviousness-type double patenting rejection.

Claim	Instant application	Claim	Co-pending application 10/306942

1	A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising: decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames; configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames; and using the image frames in the buffer memory in the decoding process.	17	(including limitation of claim 16) Claim 16 A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising: decoding, at a decoder, a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format; decoding, at the decoder, at least a second sub-sequence of the

10	A video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding, the video decoder being arranged to decode from the video sequence an indication related to at least one discontinuity of numbering of the image frames; configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames; and use the image frames in the buffer memory in the decoding process.		video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; determining, at the decoder, dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and reconstructing, at the decoder, at least part of the video sequence on the basis of the dependencies of the sub-sequences. Claim 17 A method according to claim 16, **wherein video frames are entered into a sliding buffer**

			memory in connection with decoding, further comprising: **decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional**; **configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers**; and continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to

			the missing image numbers.

(1) Regarding claim 1:

Claim 17 of the copending application discloses decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames; and configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames (the bold and underlined part in the above table).

The co-pending application does not explicitly disclose using the image frame in the buffer memory in the decoding process.

However, claim 17 also discloses wherein video frames are entered into a sliding buffer memory in connection with decoding (preamble of claim 17).

Therefore, it is obvious that the image frame in the buffer memory in the decoding process for the benefit of performing decoding with a continuous frame sequence number.

(2) Regarding claim 2:

Claim 17 further disclose wherein said indication informs that at least one discontinuity in the numbering of the image frames in the video sequence are intentional (the discontinuity in the numbering of the images of the image frames in the video sequence are intentional, claim 17).

(3) Regarding claim 3:

The co-pending application does not explicitly discloses using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

However, claim 17 discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

As the video frames are enter into a sliding buffer memory in connection with decoding, it is obvious that the comprise number of image frames corresponding to the missing image number are used in the decoding process in place of the missing frame number to provide a benefit of performing decoding with a continuous sequence number.

(4) Regarding claim 4:

Claim 17 further disclose the number of image frames in the buffer memory are used in sliding window buffering of image frame (wherein video frames are entered into a sliding buffer memory in connection with decoding).

(5) Regarding claim 5:

Clam 17 further discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames (continuing the entering of the image frames comprised by the video sequence in

question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers, claim 17).

7. Claim 6 is provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 18 of copending Application No. 10/306942.

This is a provisional obviousness-type double patenting rejection.

Claim 17 discloses all subject matter as discussed in above and claim 18 of the co-pending application depends on claim 17 and further disclose entering onto the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames (entering into the buffer memory a number of filler frames corresponding to the missing image number, claim 18).

8. Claims 10-14 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 17 of copending Application No. 10/306942.

(1) Regarding claim 10:

Claim 17 of the copending application discloses decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames; and configuring, in response to the indication, the buffer memory to provide a

number of image frames corresponding to a discontinuity in the numbering of the image frames (the bold and underlined part in the above table).

The co-pending application does not explicitly disclose (a) using the image frame in the buffer memory in the decoding process; and (b) the decoding method is performed by a decoder.

With respect to (a), claim 17 also discloses wherein video frames are entered into a sliding buffer memory in connection with decoding (preamble of claim 17).

Therefore, it is obvious that the image frame in the buffer memory in the decoding process for the benefit of performing decoding with a continuous frame sequence number.

With respect to (b), since the claim is non-structural per se, and claim 17 discloses the decoding steps, it is inherently that the decoding method is performed by a decoder.

(2) Regarding claim 11:

Claim 17 further disclose wherein said indication informs that at least one discontinuity in the numbering of the image frames in the video sequence are intentional (the discontinuity in the numbering of the images of the image frames in the video sequence are intentional, claim 17).

(3) Regarding claim 12:

The co-pending application does not explicitly discloses using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

However, claim 17 discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

As the video frames are enter into a sliding buffer memory in connection with decoding, it is obvious that the comprise number of image frames corresponding to the missing image number are used in the decoding process in place of the missing frame number to provide a benefit of performing decoding with a continuous sequence number.

(4) Regarding claim 13:

Claim 17 further disclose the number of image frames in the buffer memory are used in sliding window buffering of image frame (wherein video frames are entered into a sliding buffer memory in connection with decoding).

(5) Regarding claim 14:

Clam 17 further discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames (continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers, claim 17).

9. Claim 15 is provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 18 of copending Application No. 10/306942.

This is a provisional obviousness-type double patenting rejection.

Claim 17 of the co-pending application discloses all subject matter as discussed in above and claim 18 of the co-pending application depends on claim 17 and further disclose entering onto the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames (entering into the buffer memory a number of filler frames corresponding to the missing image number, claim 18, since the claim is non-structural per se, and claim 17 discloses the decoding steps, it is inherently that the decoding method is performed by a decoder).

10. Claims 30-31 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 17 of copending Application No. 10/306942.

(1) Regarding claim 30 (the examiner interprets the claimed signal as the received video sequence in claim 17 of the co-pending application):

Claim 17 of the copending application discloses an indication related to the discontinuity of numbering of the image frames in the transmitted video signal (decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence),

Claim 17 does not explicitly disclose the indication informs a decoder to use in a decoding process a number of image frames corresponding to the discontinuity in the numbering of the image frames.

However, claim 17 also discloses configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers, wherein video frames are entered into a sliding buffer memory in connection with decoding (preamble of claim 17).

Therefore, it is obvious that the image frame in the buffer memory in the decoding process for the benefit of performing decoding with a continuous frame sequence number.

(2) Regarding claim 31:

Claim 17 further disclose wherein said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional (the discontinuity in the numbering of the images of the image frames in the video sequence are intentional, claim 17).

11. Claims 18-22 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 17 of copending

Application No. 10/306942 in view of Langberg et al. (US 5,852,630, hereinafter Langberg).

(1) Regarding claim 18:

Claim 17 of the copending application discloses decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames; and configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames (the bold and underlined part in the above table).

The co-pending application does not explicitly disclose (a) using the image frame in the buffer memory in the decoding process; and (b) the decoding method is perform by a computer program product stored on a computer readable medium and having program code to perform the method.

With respect to (a), claim 17 also discloses wherein video frames are entered into a sliding buffer memory in connection with decoding (preamble of claim 17).

Therefore, it is obvious that the image frame in the buffer memory in the decoding process for the benefit of performing decoding with a continuous frame sequence number.

With respect to (b), Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system or

method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of claim 17 of the co-pending application would have been implemented in software. The implemented software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software in as taught by Langberg to implement the method of the co-pending application in order to reduce cost and improve the adaptability and flexibility of the communication system.

(2) Regarding claim 19:

Claim 17 further disclose wherein said indication informs that at least one discontinuity in the numbering of the image frames in the video sequence are intentional (the discontinuity in the numbering of the images of the image frames in the video sequence are intentional, claim 17).

(3) Regarding claim 20:

The co-pending application does not explicitly discloses using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

However, claim 17 discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

As the video frames are enter into a sliding buffer memory in connection with decoding, it is obvious that the comprise number of image frames corresponding to the

missing image number are used in the decoding process in place of the missing frame number to provide a benefit of performing decoding with a continuous sequence number.

(4) Regarding claim 21:

Claim 17 further disclose the number of image frames in the buffer memory are used in sliding window buffering of image frame (wherein video frames are entered into a sliding buffer memory in connection with decoding).

(5) Regarding claim 22:

Clam 17 further discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames (continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers, claim 17).

12. Claim 23 is provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 18 of copending Application No. 10/306942.

This is a provisional obviousness-type double patenting rejection.

Claim 17 and Langberg discloses all subject matter as discussed in above and Claim 18 of the co-pending application depends on claim 17 and disclose entering onto the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames (entering into the buffer memory a number of filler frames corresponding to the missing image number, claim 18, since the claim is non-structural per se, and claim 17 discloses the decoding steps, it is inherently that the decoding method is performed by a decoder).

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. Claims 26 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2).

Apostolopoulos discloses a video encoder for encoding a video sequence, the video encoder being arranged to:

encode the video sequence such that at least one image frame is exclude from the video sequence, causing a discontinuity in the numbering of image frames (encode 314 in figure 3 encode the odd frame of the original video frame 115, therefore, excluding the even frame of the original video frame 115 and causing a discontinuity in

the numbering of image frames, for example I_1, P_3, P_5, column 7, line 59 - column 8, lines 12).

Apostolopoulos does not explicitly disclose encode into the video sequence an indication related to the discontinuity of numbering of the image frames.

However, Apostolopoulos discloses that if encode block 316 generates P-frames in the encoded even frames 356 that carry motion estimation and error information according to prediction loop, then the decode block 320 uses the motion estimation and error information from the received P-frames in its own prediction loop to generate the reconstructed even video frames (column 8, lines 50-55) and encode block 314 generates the encoded odd frames 354 in response to the odd video frames 350, the encoded odd frames 354 includes I frames and P frames. In response to the example series $frame_1$, $frame_3$, $frame_5$, the encode block 314 may generate an I-frame I_1, follow by P-frames P_3, P_5, P_3 depends on I_1, and P_5 depends on P_3, column 7, lines 59-65.

Although Apostolopoulos does not explicitly disclose encode into the video sequence an indication related to the discontinuity of numbering of the image frames (missing of the even frames), it is obvious that the odd sequence would include an indication in the odd frame sequence for the dependency of P_3 to be depend on I_1, and P_5 depends on P_3, for the compensation for the missing even frames in the odd frame sequence, and also used in the decoding process.

It is desirable to encode into the video sequence an indication related to the discontinuity of numbering of the image frames because it enables diversity transmission of video stream. Therefore, it would have been obvious to one of ordinary

skill in the art at the time of invention to have an indication in the odd and even frame for the discontinuity of the numbering of frames to enable diversity transmission and improve reliability of the transmission (column 3, lines 42-44).

(2) Regarding claim 27:

Apostolopoulos further discloses that said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional (for diversity transmission of video signal, even frames and odd frames are separated into two streams, therefore, it is intentional for the separation of frames, column 7, lines 38-52).

(3) Regarding claim 28:

Apostolopoulos discloses a device for being arranged to:

receive an encoded video sequence (frame separate 312 received the original video frame 115 as shown in figure 3, column 7, lines 38-45; the examiner interprets the original video frame as a encode video sequence);

exclude at least one image frame from the video sequence, thereby causing a discontinuity in the numbering of image frames (original video sequence is separated into odd frames and even frames and forward the odd frames to encode 314 as shown in figure 3, column 3, lines 62-65).

Apostolopoulos does not explicitly disclose include into the video sequence an indication related to the discontinuity of numbering of the image frames.

However, Apostolopoulos discloses that if encode block 316 generates P-frames in the encoded even frames 356 that carry motion estimation and error information according to prediction loop, then the decode block 320 uses the motion estimation and

error information from the received P-frames in its own prediction loop to generate the reconstructed even video frames (column 8, lines 50-55) and encode block 314 generates the encoded odd frames 354 in response to the odd video frames 350, the encoded odd frames 354 includes I frames and P frames. In response to the example series $frame_1$, $frame_3$, $frame_5$, the encode block 314 may generate an I-frame I_1, follow by P-frames P_3, P_5, P_3 depends on I_1, and P_5 depends on P_3, column 7, lines 59-65.

Although Apostolopoulos does not explicitly disclose encode into the video sequence an indication related to the discontinuity of numbering of the image frames (missing of the even frames), it is obvious that the odd sequence would include an indication in the odd frame sequence for the dependency of P_3 to be depend on I_1, and P_5 depends on P_3, for the compensation for the missing even frames in the odd frame sequence, and also used in the decoding process.

It is desirable to encode into the video sequence an indication related to the discontinuity of numbering of the image frames because it enables diversity transmission of video stream. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to have an indication in the odd and even frame for the discontinuity of the numbering of frames to enable diversity transmission and improve reliability of the transmission (column 3, lines 42-44).

(4) Regarding claim 29:

Apostolopoulos further discloses that said indication informs that discontinuities in the numbering of the image frames in the video sequence are intentional (for diversity

transmission of video signal, even frames and odd frames are separated into two streams, therefore, it is intentional for the separation of frames, column 7, lines 38-52).

Allowable Subject Matter

15. Claims 7-9 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Cuccia (US 2002/0051621 A1) discloses an instant replay of digital video optimized using non MPEG tags.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU M. LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 7:30-4:00 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Examiner, Art Unit 2611
6/13/2010

/CHIEH M FAN/
Supervisory Patent Examiner, Art Unit 2611

Notice of References Cited	Application/Control No. 11/338,996	Applicant(s)/Patent Under Reexamination HANNUKSELA, MISKA	
	Examiner SIU M. LEE	Art Unit 2611	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-7,103,669	09-2006	Apostolopoulos, John G.	709/231
*	B	US-2002/0051621	05-2002	Cuccia, David W.	386/68
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	06/13/2010								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	O								
	8	O								
	9	O								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								
	19	✓								
	20	✓								
	21	✓								
	22	✓								
	23	✓								
	24	✓								
	25	✓								
	26	✓								
	27	✓								
	28	✓								
	29	✓								
	30	✓								
	31	✓								

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

SEARCHED

Class	Subclass	Date	Examiner
375	240.27, 240.25	6/14/2010	Siu M. Lee

SEARCH NOTES

Search Notes	Date	Examiner
EAST	6/14/2010	Siu M. Lee
Discuss the application with Chieh Fan	6/14/2010	Siu M. Lee

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 5361

SERIAL NUMBER	FILING or 371(c) DATE / RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/338,996	01/25/2006	375	2611	TPO.004.C1

APPLICANTS
Miska Hannuksela, Tampere, FINLAND;

**** CONTINUING DATA ***************************
This application is a CON of 10/306,942 11/29/2002

**** FOREIGN APPLICATIONS ***************************
FINLAND 20020127 01/23/2002

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****
02/28/2006

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	Met after Allowance	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
☑ Yes ☐ No	☑ Yes ☐ No	/SIU M LEE/ Examiner's Signature	☐ Initials	FINLAND	4	31	6

ADDRESS

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

TITLE

Grouping of image frames in video coding

FILING FEE RECEIVED	FEES: Authority has been given in Paper	
2150	No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	3	10/306942	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/02 13:46
S2	1	11/338996 and software	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/02 13:58
S3	1	11/338996 and (computer readable medium)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 07:09
S4	2	"5122875".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 08:05
S5	2	"5838265".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 08:08
S6	3	10/306942	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 08:20
S7	2	10/306942 and (decod$3 with buffer with memory with discontinuity)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 08:26

S8	2	"5877812".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 09:15
S9	3	10/306942 and (buffer with memory)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 09:27
S10	2	10/306942 and (buffer with memory with discontinuity)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 09:39
S11	2	10/306942 and (buffer with memory with discontinuity) and discontinuity	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 09:39
S12	2	10/306942 and decod$3 and (buffer with memory with discontinuity) and discontinuity	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 10:26
S13	49	(null or dummy or filler) with frame with (memory or buffer or storage or register) with decod $3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 14:27
S14	30	S13 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 14:28
S15	581	(frame with (identifier or indicator or tag or number$3)) with discontinu$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 15:04
S16	368	S15 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 15:04

S17	76	S16 and ((buffer or memory or register or stor$3) with decoding)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 15:05
S19	31	S17 and video	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 15:09
S20	20	number$3 with ((video or image) frame) with discontinu$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:34
S21	0	discard$3 with (image or video) frame with discontinu$4 with number$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:55
S22	0	discard$3 with (image or video) frame with discontinu$4 with label$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:55
S23	0	discard$3 with (image or video) frame with discontinu$4 with tag	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:55
S24	62	discard$3 with (image or video) frame with number$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:55
S25	33	S24 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:56
S26	2636	375/240.27,240.25.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/13 20:57

S27	1417	S26 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/13 20:57
S28	49	S27 and ((discontinu$4 or miss $3 or skip$4) with frame with number$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/13 20:59

EAST Search History (Interference)

< This search history is empty>

6/14/2010 8:22:24 AM
C:\Documents and Settings\slee7\My Documents\EAST\Workspaces\11338996 201006.wsp

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	11-177987	02.07.1999	JP			Abstract	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	2010	English Translation of the Office Action of parallel Japanese application no. 2006-120296 dated 01/06/2010.

EXAMINER /Siu Lee/	DATE CONSIDERED 06/02/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/SL/	6496980	12/17/2002	Tillman et al.			
	6510553	01/21/2003	Hazra			
	6639943	10/28/2003	Radha et al.			
/SL/	6614936	09/02/2003	Wu et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	EP0898427	2/24/1999	EP			NA	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	5/7/1996	Elloumi et al., "Issues in Multimedia Scaling: the MPEG Video Streams Case", Emerging Technologies and Applications in Communications, IEEE Computer Society, May 7, 1996.
	2/1/2002	Hannuksela, "Enhanced Concept of GOP", Joint Video Team of ISO-IES MPEG 7 itu-T VCEG, 2/1/2002.
	10/12/1997	Soderquist et al., "Memory Traffic and Data Cache Behavior of an MPEG-2 Software Decoder", IEEE Computer Society, October 12, 1997.
	2/2/2010	Copy of Office Action of related European application no. 06101013.8 dated 02/02/2010.
	2/2/2010	Copy of Office Action of related European application no. 06101014.6 dated 02/02/2010.
	3/16/2006	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 3/16/2006, 4 pages.
	8/10/2006	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 8/10/2006, 16 pages.
	9/11/2007	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 9/11/2007, 4 pages.
	3/13/2007	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 3/13/2007, 7 pages.
/SL/	12/7/2009	Copy of Notice of Allowance of parallel U.S. Application No. 10/306,942 dated 12/7/2009, 8 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 06/02/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

/SL/	5/6/2006	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 8/6/2006, 13 pages.
	11/9/2006	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 11/9/2006, 16 pages.
	5/31/2007	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 5/31/2007, 12 pages.
/SL/	7/9/2009	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 7/9/2009, 12 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 06/02/2010
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	EP 0798932	10/1/1997	EP			NA	
/SL/	CN 1173783A	2/18/1998	China				X

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER /Siu Lee/	DATE CONSIDERED 06/02/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/SL/	5122875	06/16/1992	Raychaudhuri et al.			
/SL/	5838265	11/17/1998	Adolph			
/SL/	5877812	03/02/1999	Krause et al.			
/SL/	6314139	11/06/2001	Koto et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	02/04/2010	Copy of Allowance Notification with translation dated 2/4/2010 from parallel Russian Application No. 2006110322, 17 pages.
/SL/	02/25/2010	Copy of translated Office Action dated 2/25/2010 from parallel Japanese Application No. 2006-120297, 5 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 06/02/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/SL/	6,266,158	7/24/2001	Hata et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	10-271511	10/09/1998	JP				X

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER /Siu Lee/	DATE CONSIDERED 06/02/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Date Mailed: June 10, 2008

OIPE IAP76 JUN 16 2008 PATENT & TRADEMARK OFFICE

FORM 1449*	INFORMATION DISCLOSURE STATEMENT	Docket Number: TPO.004.C1	Application Number: 11/338,996
	IN AN APPLICATION	Applicant: HANNUKSELA	
	(Use several sheets if necessary)	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/SL/	20010024472	09/27/2001	Sporer et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	/Siu Lee/	DATE CONSIDERED	06/02/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

OIPE IAP2 JUN 19 2006 PATENT & TRADEMARK OFFICE

FORM 1449	INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
		Applicant: HANNUKSELA	
		Filing Date: 01/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/SL/	5,699,476	16.12.1997	VAN DER MEER			
	5,774,593	30.6.1998	ZICK ET AL.			
	6,108,382	22.8.2000	GRINGERI ET AL.			
	6,307,886	23.10.2001	WESTERMANN			
/SL/	6,337,881	8.1.2002	CHADDHA			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	WO 00/67469	9.11.2000	PCT			n/a	
	WO 01/47283	28.06.2001	PCT			n/a	
	WO 02/054776	11.7.2002	PCT			n/a	
/SL/	EP 1 005 232	31.05.2000	EP			n/a	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	9-17 Oct. 2002	Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, *Editor's Proposed Draft Text Modification for Joint Video Specification (ITU-T Rec. 264 ISO/IEC 14496-10 AVC)*, Geneva modifications draft 37.
	15 Jan. 1998	ITU-Telecommunications Standardization Union, *Draft Text of Recommendation H.263 Version 2 ("H.263+") for Decision.*
	7/10/02	ITU-Telecommunications Standardization Union, *H.26L Test Model Long Term Number 8 (TML-8) Draft0*, Document VCEG-N10. pp. 1-46.
	Nov. 2000	ITU-Telecommunications Standardization Union, *Draft for "H.263++" Annexes U, V, and W to Recommendation H.263.*
	12 Nov. 2002	David Singer and Toby Walker, *Study Text of ISO/IEC 14496-15/FCD,* pp. 1-25.
	12 Nov. 2002	David Singer and Toby Walker, *Study Text of ISO/IEC 14496-15/FCD (Revised),* 26 pages.
	Sept. 2002	Hannuksela, M et al.; Sub-Picture: ROI Coding and Unequal Error Protection, IEEE 2002 Conference, September 2002, USA (internet publication).
	Sept. 2002	Hannuksela, M et al.; Sub-Picture: Video-Coding for Unequal Error Protection, XI European Signal Processing Conference, September 2002, Toulouse, France (internet publication).
	7.12.2001	Wang, Y.K. et al.; Core Experiment Description of Sub-Picture Coding, ITU 15. Meeting 3.- 7.12.2001, Pattaya, Thailand (internet publication)
/SL/	2001	B.A. Lokshin, "Digital Broadcasting: from Studio to Audience" Cyrus Systems, Moscow, 2001. (translation included)

EXAMINER /Siu Lee/	DATE CONSIDERED 06/02/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

OIP IAP19 AUG 18 2006 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	1998	Wenger, "Temporal Scalability Using P-pictures for Low-latency Applications", IEEE Workshop, December 1998, pp. 559-564.

EXAMINER	/Siu Lee/	DATE CONSIDERED	06/02/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

OIPE OCT 22 2007 PATENT & TRADEMARK OFFICE

FORM 1449* (modified)	INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
		Applicant: HANNUKSELA	
		Filing Date: Jan. 25, 2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/SL/	5,144,426	09.01.1992	Tanaka et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	01/84850	8 Nov. 2001	WO			NA	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	/Siu Lee/	DATE CONSIDERED	06/02/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

FORM 1449	INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
		Applicant: HANNUKSELA	
		Filing Date: 1/25/2006	Group Art Unit: 2613

OIPE IAP43 NOV 24 2006 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	RU 2 123 769	20.12.1998	Russia			Abstract Only	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	/Siu Lee/	DATE CONSIDERED	06/02/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	4390966	06/28/1983	Kawashima et al.			
	5852630	12/22/1998	Langberg			
	6072831	06/06/2000	Chen			
	6094456	07/25/2000	Ueda			
	6363208	03/26/2002	Nitta et al.			
	20010040700	11/15/2001	Hannuksela et al.			
	20030063806	04/03/2003	Kim et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	2137197C1	9/10/1999	Russia				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	1997	Illgner et al., "Spatially Scalable Video Compression Employing Resolution Pyramids", IEEE Journal on Selected Areas in Communications, Vol. 15, No. 9, December 1, 2007, pg. 1688-1703.
	2000	Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Systems, ISO/IEC 13818-1, Second edition, December 1, 2000, 174 pages.
	5/5/2010	Notice of Allowance dated 5/5/2010 from Russian Application No. 2006110321/09, 11 pages.
	5/25/2010	Office Action dated 5/25/2010 from U.S. Serial No. 11/338,934, 29 pages.
	6/23/2010	Office Action dated 6/23/2010 from U.S. Serial No. 10/306,942, 6 pages.
	5/25/2010	Office Action dated 5/25/2010 from U.S. Serial No. 10/306,942, 6 pages.
	4/09/2009	Office Action dated 4/9/2009 from U.S. Serial No. 10/306,942, 4 pages.
	5/8/2006	Office Action Response dated 5/8/2006 from U.S. Serial No. 10/306,942, 11 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	TAPA.004PA	3642

76385 7590 06/23/2010

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
DIEP, NHON THANH	
ART UNIT	PAPER NUMBER
2621	
MAIL DATE	DELIVERY MODE
06/23/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/306,942	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	Nhon T. Diep	2621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 March 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13,15-24,26 and 27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-13,15,16,19,20,22,24,26 and 27* is/are rejected.

7)☒ Claim(s) *17,18 and 21* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 December 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 112

1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claims 2-10 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Claim 2 recites “encoding said interdependency into the video sequence”, and because the claimed subject matter was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention, Therefore the claims fail to comply with 35 U.S.C. 112, first paragraph.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1, 11-13, 15-16, 19-20, 22-24 and 26-27 are rejected under 35 U.S.C. 102(b) as being anticipated by Nitta et al (US 6,363,208 B2).

Nitta et al discloses a digital signal reproduction method and apparatus comprising the same method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format (fig. 2A, I = Intra coding; and P = inter coding), the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames (1st format: I and/or P; 2nd format P and/or B), the method comprising:

encoding, at an encoder, the video sequence as at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format (1ST frame of the GOP is intra-frame coded of fig. 2A);

encoding, at the encoder, the video sequence as at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence(2 nd frame of the GOP is inter-frame coded); and

encoding, at the encoder, into the video sequence information indicating which video frames belong to the second sub-sequence (any inter-coded frame (P or B) is considered belonging to the second sub-sequence and col. 1, ln. 44-56) as specified in claims 1, 15-16, 22-24 and 26; wherein the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference

image (MPEG-2 GOP, I, P, B frames) as specified in claim 11; further comprising: coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping (Forward and/or backward P frame(s) and B frame(s)) as specified in claim 12; further comprising: coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific (MPEG-2 Predictive Coding uses 8 X 8 macroblock) as specified in claim 13; further comprising: initiating the decoding from a random point in the video sequence (beginning of any GOP); determining an independently decodable sub-sequence that is next after the random point (Intra-coded frame and/or inter-coded frames), and setting the value of the image number of the first video frame in the sub-sequence at zero (the 1st frame of the GOP is identified as I sub 0) as specified in claim 20; and a removal of the second subsequence from the bitstream does not prevent decoding of the bitstream correctly (removal of P or B frame would not cause the incorrect decoding of I frame) as specified in claims 19 and 27.

Allowable Subject Matter

5. Claims 17-18 and 21 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nhon T. Diep whose telephone number is 571-272-7328. The examiner can normally be reached on m-f.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mehrdad Dastouri can be reached on 571-272-7418. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Nhon T Diep/

Primary Examiner, Art Unit 2621

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	TAPA.004PA	3642

76385 7590 05/25/2010

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
DIEP, NHON THANH	
ART UNIT	PAPER NUMBER
2621	

MAIL DATE	DELIVERY MODE
05/25/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/306,942	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	Nhon T. Diep	2621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 March 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13,15-24,26 and 27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-13,15-16, 19-20, 22-24, 26 and 27* is/are rejected.

7)☒ Claim(s) *17,18 and 21* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 December 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 112

1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claims 2-10 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Claim 2 recites "encoding said interdependency into the video sequence", and because the claimed subject matter was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention, Therefore the claims fail to comply with 35 U.S.C. 112, first paragraph.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1, 11-13, 15-16, 19-20, 22-24 and 26-27 are rejected under 35 U.S.C. 102(b) as being anticipated by MPEG-2 Predictive Coding Protocol.

MPEG-2 Predictive Coding discloses the same method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format (MPEG-2 Predictive Coding, Intra and inter coding), the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames (1st format: I and/or P; 2nd format P and/or B), the method comprising:

encoding, at an encoder, the video sequence as at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format (1ST frame of the GOP is intra-frame coded);

encoding, at the encoder, the video sequence as at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence(2 nd frame of the GOP is inter-frame coded); and

encoding, at the encoder, into the video sequence information indicating which video frames belong to the second sub-sequence (any inter-coded frame (P or B) is considered belonging to the second sub-sequence) as specified in claims 1, 15-16, 22-24 and 26; wherein the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image (MPEG-2 GOP, I, P, B frames) as specified in claim 11; further comprising: coding the sub-sequences in

such a way that at least some of the sub-sequence frames are temporally at least partly overlapping (Forward and/or backward P frame(s) and B frame(s)) as specified in claim 12; further comprising: coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific (MPEG-2 Predictive Coding uses 8 X 8 macroblock) as specified in claim 13; further comprising: initiating the decoding from a random point in the video sequence (beginning of any GOP); determining an independently decodable sub-sequence that is next after the random point (Intra-coded frame and/or inter-coded frames), and

setting the value of the image number of the first video frame in the sub-sequence at zero (the 1st frame of the GOP is identified as Io) as specified in claim 20; and a removal of the second subsequence from the bitstream does not prevent decoding of the bitstream correctly (removal of P or B frame would not cause the incorrect decoding of I frame) as specified in claims 19 and 27.

Allowable Subject Matter

5. Claims 17-18 and 21 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Lyons et al (US 6,330,286) discloses a flow control, latency control, and bitrate conversions in a timing correction and frame synchronization apparatus.

b. Radha et al (US 6,806,909) discloses seamless splicing of MPEG-2 multimedia data streams.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nhon T. Diep whose telephone number is 571-272-7328. The examiner can normally be reached on m-f.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mehrdad Dastouri can be reached on 571-272-7418. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Nhon T Diep/
Primary Examiner, Art Unit 2621

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/306,942	11/29/2002	Miska Hannuksela	TAPA.004PA	3642

76385 7590 04/09/2009

Hollingsworth & Funk, LLC
8009 34th Avenue South
Suite 125
Minneapolis, MN 54425

EXAMINER	
DIEP, NHON THANH	
ART UNIT	**PAPER NUMBER**
2621	

MAIL DATE	DELIVERY MODE
04/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/306,942	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	Nhon T. Diep	2621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 March 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13,15-24,26 and 27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *15 and 21-24* is/are allowed.

6)☒ Claim(s) *1-13,16-21,26 and 27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *11/29/2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claim 14 is cancelled as per paragraph of the previous Office Action mailed out on 9/11/2007.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1-13, 16-21 and 26-27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Regarding to claims 1-13, 16-21 and 26-27, which are method claims, and that the claims do not fall within one of the four statutory categories of inventions. Supreme Court precedent and recent Federal Circuit decisions indicate a statutory "process" under 35 U.S.C. 101 must (1) be tied to another statutory category (such as a particular apparatus), or (2) transform underlying subject matter (such as an article or material) to a different state or thing. While the instant claim(s) recite a series of steps or acts to be performed, it is considered that none of the steps in the claims positively recites the other statutory class (the thing or product) to which it is tied, for example, by identifying the apparatus that accomplishes the method steps, In re Bilski, 88 USPQ2d 1385 (Fed. Cir. 2008).

Allowable Subject Matter

4. Claims 15, 22-24 are allowed.

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nhon T. Diep whose telephone number is 571-272-7328. The examiner can normally be reached on m-f.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mehrdad Dastouri can be reached on 571-272-7418. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ND

/Nhon T Diep/
Primary Examiner, Art Unit 2621

OIPE IAP21 MAY 08 2006 PATENT & TRADEMARK OFFICE

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Diep, N.
Serial No.:	10/306,942	Group Art Unit:	2613
Filed:	November 29, 2002	Docket No.:	TAPA.004PA
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service with sufficient postage, as first class mail, in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on May 5, 2006.

By Tracey M. Dotter

RESPONSE TO RESTRICTION REQUIREMENT AND PRELIMINARY AMENDMENT

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Examiner's requirement for restriction dated March 16, 2006, the claims of Group I (Claims 1-13 and 15-25) are provisionally elected *with traverse*. The Applicant respectfully requests reconsideration and withdrawal of the requirement for restriction, for reasons set forth below. The Applicant respectfully requests reconsideration and allowance of all pending claims.

Claims 26 and 27 have been added. Authorization is given to charge Deposit Account No. 50-3581 (TAPA.004PA) in the amount of $300.00 for the two additional claims added by this filing. Authorization is also given to charge/credit Deposit Account No. 50-3581 (TAPA.004PA) for any deficiencies/overpayments.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 10 of this paper.

Listing of Claims:

1. (Original) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

forming into the video sequence information indicating which video frames belong to the second sub-sequence.

2. (Original) A method according to claim 1, further comprising

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

3. (Original) A method according to claim 2, further comprising

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from a group comprising

- a video frame of a higher scalability layer,
- a video frame of another sub-sequence in the same scalability layer.

4. (Original) A method according to claim 3, further comprising

determining the dependencies of the video frames of the second sub-sequence on the basis of at least a scalability layer identifier and a sub-sequence identifier.

5. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

6. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

7. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.

8. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame on the basis of the image number.

9. (Original) A method according to claim 7, further comprising

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. (Original) A method according to claim 7, further comprising

adding the identifier to the Supplemental Enhancement Information (SEI) data structure to be transmitted synchronously with the video sequence.

11. (Original) A method according to claim 1, wherein

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

12. (Original) A method according to claim 1, further comprising

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

13. (Original) A method according to claim 1, further comprising

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

14. (Original) An element in a streaming system, such as a streaming server or a network element in a telecommunications system, wherein said element is arranged to

transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission,

control the bit rate of a video sequence transmission in which a first sub-sequence of the video sequence comprises video frames of at least a first frame format and at least a second sub-sequence comprises video frames of at least a second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

the bit rate control being arranged to be carried out in at least one of the following ways:

- at least one sub-sequence is removed from the video sequence,
- at least one sub-sequence is added to the video sequence.

15. (Original) A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format; and to

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame

of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and to

form into the video sequence information indicating which video frames belong the second sub-sequence.

16. (Original) A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

decoding a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;

decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determining dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and

reconstructing at least part of the video sequence on the basis of the dependencies of the sub-sequences.

17. (Original) A method according to claim 16, wherein video frames are entered into a sliding buffer memory in connection with decoding, further comprising

decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;

configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

18. (Original) A method according to claim 17, further comprising

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

19. (Original) A method according to claim 16, further comprising

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

20. (Original) A method according to claim 16, further comprising

initiating the decoding from a random point in the video sequence;

determining an independently decodable sub-sequence that is next after the random point, and

setting the value of the image number of the first video frame in the sub-sequence at zero.

21. (Original) A method according to claim 16, further comprising

identifying at least partly temporally overlapping sub-sequence frames from the video sequence based on the picture order count information, and

outputting from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.

22. (Original) A video decoder for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video decoder is arranged to

decode a first sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the first frame format;

decode at least a second sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video

frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.

23. (Original) A computer program for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining data into the video sequence to determine which video frames belong to the second sub-sequence.

24. (Original) A computer program for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second

frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.

25. (Original) A video signal comprising a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video signal comprises

a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format;

at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

at least one data field determining the video frames that belong to the second sub-sequence.

26. (New) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format;

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

forming into the video sequence information indicating which video frames belong to the second sub-sequence, wherein subsequence information is different from picture type information.

27. (New) A method according to claim 26, wherein

a removal of the second subsequence from the bitstream does not prevent decoding of the bitstream correctly.

REMARKS/ARGUMENTS

The election/restriction requirement dated March 16, 2006, has been considered. The Examiner contends that the claims are drawn to two inventions, namely Group I (Claims 1-13 and 15-25) and Group II (Claim 14) and asserts that restriction to one of claim Groups I-II is required under 35 U.S.C. §121. The claims of **Group I (Claims 1-13 and 15-25)** are provisionally elected *with traverse*. The Applicant traverses the restriction requirement as it fails to satisfy the requirements for maintaining such a restriction requirement and accordingly requests reconsideration and withdrawal of the restriction requirement.

The election/restriction requirement asserts that the claims of Groups I and II are unrelated. It appears that the Examiner is asserting that the claims of the two Groups are independent based upon the reliance on §§802.01 and 806.06 of the MPEP. Specifically, §806.06 indicates that two inventions are independent if there is no disclosed relationship between the inventions, that is, they are unconnected in design, operation, and effect. The standard for showing that two inventions are independent is difficult to prove, for example, a specific process of molding would be independent from a molding apparatus so long as the apparatus cannot be used to practice the specific process. The Examiner's assertion that the claims of Group I are directed to forming a scalable, compressed video sequence, whereas the claim of Group II is directed to control of the bit rate of the compressed video packets for transmission, fails to identify how the two alleged inventions are disclosed as not capable of use together. The Examiner fails to provide any evidence to support the conclusion that the two alleged inventions are independent. Without any such evidence, the restriction requirement is improper. Accordingly, the Applicant requests that the restriction requirement be withdrawn.

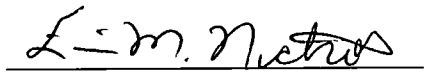
New Claims 26 and 27 have been added. Claim 26 largely replicates Claim 1 and is believed to be directed to a similar invention if the above election/restriction requirement is maintained; therefore, the new claims are believed to be associated with the Examiner's asserted Group I.

CONCLUSION

In view of the above, the Applicant respectfully requests reconsideration and withdrawal of the requirement for restriction. If the Examiner would find it helpful to discuss this issue by telephone, the undersigned attorney of record invites the Examiner to contact the attorney of record.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

By: 
Erin M. Nichols
Reg. No. 57,125

Spatially Scalable Video Compression Employing Resolution Pyramids

Klaus Illgner and Frank Müller

Abstract— **In this paper, a spatially scalable video coding scheme for low bit rates is proposed. The codec is especially well suited for communications applications because it is based on motion-compensated predictive coding which provides the necessary low-delay property. The frames to be coded are decomposed into a Gaussian pyramid. Motion estimation and compensation are performed between corresponding pyramid levels of successive frames. We show that, to fulfill specific needs of spatial scalability, the motion compensation on each level must result in compatible prediction errors (displaced frame differences, DFD). Compatibility of the prediction errors means that the pyramid formed by independently obtained DFD's (the DFD pyramid) is close to a Gaussian pyramid decomposition of the DFD of the highest resolution level. From the DFD pyramid, a least squares Laplacian pyramid is derived, which is quantized and coded. The DFD encoder outputs an embedded bit stream. Thus, the coder control may truncate the bit stream at any point, and can keep a fixed rate. The motion vector fields obtained at the different resolution levels are also encoded by employing a pyramid approach. Simulation results show that the proposed coder achieves a coding gain compared to simulcast coding.**

Index Terms—**Laplacian pyramid, multiresolution pyramid, scalable motion compensation, spline filter, video compression.**

I. Introduction

THE rapid development of networks leads to increasing interest in video communications. All current standards for communications-related video compression (H.261, H.263) are based on a single resolution hybrid (i.e., motion-compensated predictive) coding scheme. However, bandwidth limitations, multipoint operation with receivers of different capabilities, and bandwidth-dependent charging are good reasons to use scalable coding schemes. Also, MPEG-4 announced strong demands for scalable video coding algorithms.

The term *scalability* is ambiguous because it is used in a spatial, temporal, or SNR context. In this paper, the focus is on spatial multiresolution schemes since this is an essential prerequisite to provide the property of *spatial* scalability. This means that the receiver can reconstruct frames of smaller spatial resolution using only a subset of the complete bit stream, thus fulfilling the above-mentioned demands.

In search of an appropriate coding principle, spatiotemporal subband or wavelet approaches are problematic because the coding delay must be kept as small as possible for communications applications. A promising approach is therefore to extend the classical hybrid coding principle, which is the basis of most coding schemes for communications, to fulfill the needs of spatial scalability.

Known multiresolution approaches based on the hybrid coding principle utilize several DPCM loops on different resolution levels. The design objective of a spatially scalable video codec is to find optimal predictions for each resolution level. However, this task cannot be solved straightforwardly because the prediction errors have to be coded efficiently. This can be accomplished only by exploiting the dependencies between the different resolution levels. The degree of coupling between the DPCM loops makes for the main difference between various schemes.

A straightforward approach is to decompose the frames into multiple resolution frames, and to code each resolution level independently. However, this approach results in significant coding overhead. The scalable codec described in [1] is based on MPEG-II, and employs several DPCM loops on different resolution levels. There exists a slight interconnection between the layers since motion estimation is performed in a hierarchical fashion.

A tighter interconnection is realized in [2], where a two-layer scalable pyramid codec based on two DPCM loops is described. Hierarchical motion estimation is employed in combination with a hierarchical VQ on the displaced frame differences. There exist also various two-dimensional (2-D) wavelet-based and subband schemes [3]–[5]. Motion estimation and compensation turn out to be difficult in the wavelet/subband domain. One difficulty arises from the shift variance of downsampling in the wavelet domain [3]. Another reason is decreased efficiency of motion compensation on bandpass frames since the high-pass signal components are disturbed due to noise and aliasing. Therefore, most approaches perform motion compensation on low-pass subbands.

In this paper, a hybrid spatially scalable video coding algorithm is presented which is based on a Gaussian pyramid decomposition. Motion estimation and compensation are performed between each pyramid level such that the resulting prediction errors are close to a Gaussian pyramid decomposition of the prediction error at the highest resolution level. For coding of the displaced frame differences (DFD) as well as the motion vector fields, the statistical dependencies between the pyramid levels (corresponding to the different resolutions) are utilized by coding jointly all resolutions with a compact Laplacian pyramid representation. More specifically, the DFD is decomposed into a centered Laplacian least squares spline pyramid (construction and properties of this kind of pyramid are derived in this paper), and subsequently coded in an embedded fashion.

Manuscript received September 9, 1996; revised July 2, 1997.

The authors are with the Institut für Elektrische Nachrichtentechnik, RWTH Aachen, 52056 Aachen, Germany.

Publisher Item Identifier S 0733-8716(97)07705-6.

0733–8716/97$10.00 © 1997 IEEE

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

One of the most efficient embedded image coding algorithms has been developed by Shapiro [6], and is called the embedded zero-tree wavelet (EZW) coding algorithm. It consists of a transformation of the image into a wavelet domain, and subsequent embedded coding with the zero-tree algorithm, which starts with coarse quantizations and refines the wavelet coefficients in subsequent passes. Taubman [7] developed a three-dimensional (3-D) subband video coder, which (restricted to 2-D) resembles the EZW coder in the aspects of decomposition and layered quantization. Instead of using zero-trees for coding of significance maps, he used a conditioning context to predict zeros in dominant passes. He shows that this kind of general prediction context provides significant gain over the zero-tree coding method. In both cases, a wavelet decomposition has been used. In a previous work [8], we adapted the EZW coding scheme to a Laplacian pyramid decomposition, and discussed the usage of conditional arithmetic coding in a Laplacian pyramid. We showed there that the conditional arithmetic coder outperforms the pyramid zero-tree coder.

In this paper, we adapt the Laplacian pyramid coding of displaced frame differences to a spatially scalable coding scheme. We have to deal with two DFD's per frame, corresponding to the base layer and the refinement layer of the codec. The DPCM loops for both layers are coupled such that the DFD in the base layer is "close" to a reduced version of the DFD of the refinement layer. This requires a careful design of the filter operators which accomplish the transition between the different resolutions. Requirements for these filters are given in this paper, together with a new filter design which fulfills these requirements.

The paper is organized as follows. In the next section, an overview of the hybrid spatially scalable coding concept is given. The third section discusses motion compensation for scalable video coding in general, and mentions particular implementation issues of the proposed scheme. In the fourth section, desirable properties for the reduce and expand operators (which define the pyramid decompositions) are summarized. While most of these properties can be fulfilled with any least squares pyramid [9], employment in the scalable codec makes the *centering* a desired feature. The derivation of such pyramids is included in the Appendix. The fifth and sixth sections describe DFD coding and vector field coding in detail, and the last two sections are devoted to simulation results and some conclusions.

II. Design of a Spatially Scalable Coding Scheme

We start with a discussion of the motion-compensated predictive coder shown in Fig. 1. The current frame is denoted by g_n and the prediction of g_n by $\hat{g}_n$. The DFD $d_n = g_n - \hat{g}_n$ is decomposed into a Gaussian pyramid. A Laplacian pyramid is derived from the Gaussian pyramid and coded in embedded fashion using arithmetic coding. The prediction $\hat{g}_n$ is calculated as a motion-compensated version of the previously reconstructed frame $\tilde{g}_{n-1}$ using overlapping blocks. For motion estimation between the frames g_n and $\tilde{g}_{n-1}$, a gain–cost criterion is employed. The motion vector field v_n is

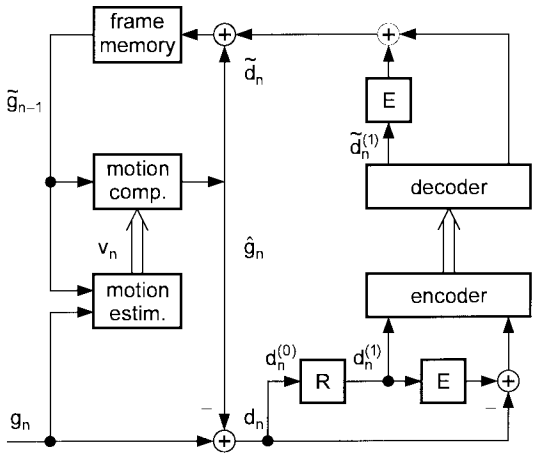


Fig. 1. Block diagram of *PYRACO*, a predictive multiresolution video coding scheme (nonscalable version).

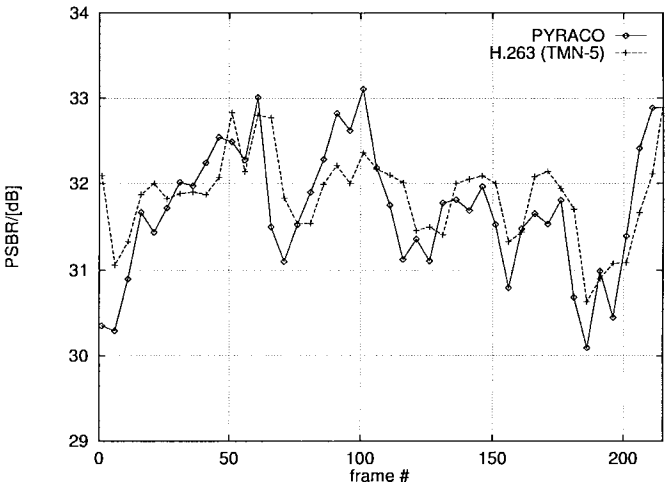


Fig. 2. PSNR of sequence *foreman* coded at 48 kbit/s and 5 fps (fixed).

coded losslessly in a hierarchical fashion [10]. Fig. 2 shows[1] that this coding approach, which is termed PYRACO, reaches approximately the same coding performance as the currently best reference coder H.263 (TMN-5 implementation) [11]. An important difference is that PYRACO outputs a constant bit rate per frame, which simplifies coder control and reduces coding delay, while TMN generates a significantly varying rate.

The multiresolution image representation as a Gaussian pyramid offers a natural design choice for a spatially scalable coding scheme with multiple resolution levels. For simplicity of presentation, we consider in the following only two resolution levels. A block diagram of the spatially scalable hybrid encoder is depicted in Fig. 3. The *base layer* coder emits the lower spatial resolution, and follows the same coding principle as the coder in Fig. 1. A temporally predictive coding approach is also used for encoding of the *refinement layer*, which provides the higher spatial resolution.

[1]The test sequence has been obtained from the CCIR sequence using spline filters [9] for preserving sharpness. The TMN codec runs with all options except PB frames.

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

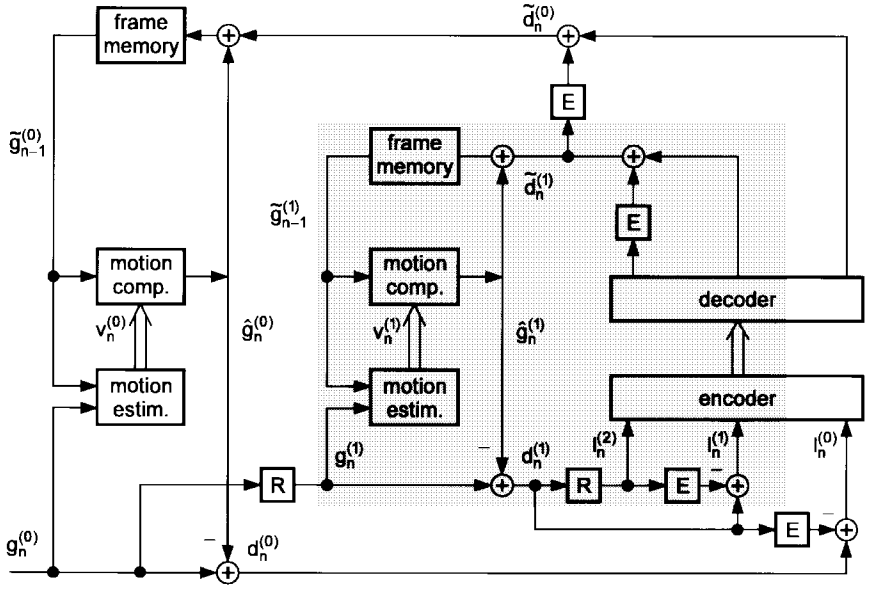


Fig. 3. Encoder of a two-level scalable predictive video coding scheme (the base layer is shaded).

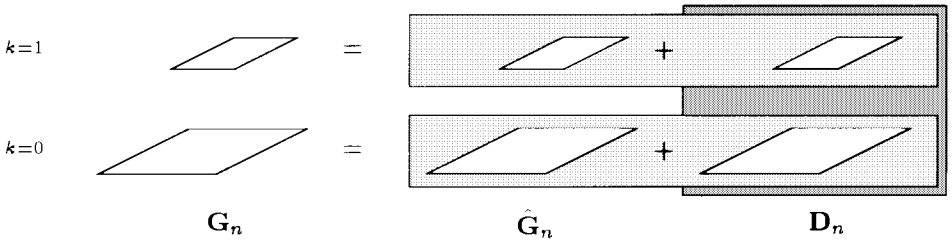


Fig. 4. Relations between two resolutions in a predictive video coding scheme.

A. Aim for Motion Compensation to Achieve Scalability

The first step is a decomposition of the current frame g_n into a Gaussian pyramid $\boldsymbol{G}_n$ of two levels[2]

$$\boldsymbol{G}_n = (g_n^{(0)}, g_n^{(1)}) = (g_n, R(g_n)). \tag{1}$$

$R(\cdot)$ denotes the so-called reduce operator, which consists of a linear low-pass filter followed by subsampling of the image in each dimension by a factor of 2. The aim is now to employ a predictive video compression scheme for both resolution levels of the frame to be coded. Hence, both levels are decomposed into a predicted frame and the prediction error, which is depicted in Fig. 4 reading the figure horizontally. However, reading Fig. 4 vertically, the prediction errors of both resolution levels form a pyramid. This pyramid can be coded efficiently [8] if it is a Gaussian-type pyramid.

Hence, the idea for the design of a spatially scalable predictive video compression scheme is to calculate predictions $\hat{g}_n^{(k)}$ for each level k

$$d_n^{(k)} = g_n^{(k)} - \hat{g}_n^{(k)}, \qquad k = 0, 1$$

such that the relation

$$R\left(d_n^{(0)}\right) \stackrel{!}{=} g_n^{(1)} - \hat{g}_n^{(1)} \tag{2}$$

holds, which is equivalent to the condition

$$R\left(\hat{g}_n^{(0)}\right) \stackrel{!}{=} \hat{g}_n^{(1)}. \tag{3}$$

The predicted levels are obtained by motion compensation of the corresponding level of the previous frame:

$$\hat{g}_n^{(k)} = \mathrm{MC}\left(\tilde{g}_{n-1}^{(k)}, v_n^{(k)}\right).$$

If the predictions satisfy (3), the prediction errors in Fig. 4 indeed constitute a Gaussian pyramid $\boldsymbol{D}_n$.

In a spatially scalable coding scheme, the prediction for the base layer needs to be calculated *without* considering the higher resolution level. This constraint is the central difficulty in the design.

Since the operator $\mathrm{MC}(\cdot)$ is nonlinear and space variant, motion compensation and low-pass filtering cannot be interchanged in general. Furthermore, subsampling is a shift-variant operation causing additional aliasing errors. Therefore, the low-pass filtered prediction from level $k = 0$ differs from the prediction on level $k = 1$, even if the motion is known exactly at level 0 [13]. Hence, equality in (3) cannot be obtained in general.

In order to achieve high compression efficiency, the aim is to find predictors for each scalable level such that (3) is approximated sufficiently accurate

$$R\left(\hat{g}_n^{(0)}\right) \approx \hat{g}_n^{(1)}. \tag{4}$$

[2] Throughout the paper, we use the convention that level 0 corresponds to the refinement layer, and level 1 to the base layer.

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

The design objective is to make the prediction of the base layer similar to the *reduced* prediction of the higher resolution level. The approach followed in this paper calculates the base layer prediction on an expanded frame, and subsequently applies a reduce operator. It is shown in Section III that by appropriately selecting the expand and reduce filters, (4) can be achieved sufficiently accurate. It turns out that the *same* filters as for the DFD pyramid coding are suitable.

B. Pyramid Decomposition of the Base Layer DFD

The base layer DFD $d^{(1)}$ is decomposed into a Laplacian pyramid using centered cubic splines (see Appendix B), which results in high energy concentration into the higher pyramid levels. The decomposition follows the *Gaussian–Laplacian* pyramid approach [12]. First, an intermediate pyramid structure, the Gaussian DFD pyramid $\boldsymbol{D}'$, is generated. The pyramid is initialized at the lowest level $l = 1$ with the original base layer DFD we want to code. From this level, a coarse approximation is derived by applying a reduce operator $R(\cdot)$. This procedure is iterated, resulting in the base layer Gaussian pyramid

$$\boldsymbol{D}' = \left\{ \left(d^{(1)}, \cdots, d^{(L-1)} \right) | d^{(k)} = R\left(d^{(k-1)} \right), \quad k = 2, \cdots, L-1 \right\} \tag{5}$$

consisting of $L-1$ levels.[3]

For computation of the corresponding Laplacian pyramid, an expand operator $E(\cdot)$ is defined, which predicts a level of the Gaussian pyramid from the next higher level, thus expanding the size again by a factor of 2 for each dimension. The expand operator consists of upsampling and subsequent filtering with a linear interpolation filter. The Laplacian pyramid $\boldsymbol{L}'$ captures the loss of information occurring through the reduction of the spatial resolution by means of the difference between a Gaussian image at level k and the expanded version of the Gaussian image at level $k+1$:

$$\boldsymbol{L}' = \left(l^{(1)}, l^{(2)}, \cdots, l^{(L-1)} \right) \tag{6}$$

where

$$l^{(k)} = d^{(k)} - E\left(d^{(k+1)} \right), \qquad k = 1, \cdots, L-2.$$

The top level of $\boldsymbol{L}'$ is "initialized" with the top level of $\boldsymbol{D}'$

$$l^{(L-1)} = d^{(L-1)}.$$

Knowledge of the Laplacian pyramid is sufficient to reconstruct the base layer DFD $d^{(1)}$. Thus, an encoder can build up a coarse representation and successive refinements (the Laplacian levels) of an image, and a decoder can reconstruct the original from this information.

[3]According to our convention, the complete pyramid consists of L levels, and therefore the base layer pyramid has $L-1$ levels.

C. Approach for DFD and Vector Field Coding

As is depicted in the block diagram of the spatially scalable hybrid encoder in Fig. 3, motion estimation and compensation are performed on the resolution levels 0 and 1, yielding a prediction error for the base and the refinement layer. The base layer prediction error is decomposed further into a Gaussian pyramid according to (5). Because the receiver needs only to know about the expand operator, the levels of the pyramid can be decomposed by employing different reduce operators, which may even be nonlinear. Therefore, the complete Gaussian pyramid of the prediction errors $\boldsymbol{D}$ is obtained by concatenating the prediction error of the refinement layer and the Gaussian pyramid of the base layer

$$\boldsymbol{D} = (d^{(0)}, \boldsymbol{D}'). \tag{7}$$

From this pyramid, the *complete* Laplacian pyramid

$$\boldsymbol{L} = \left\{ \left(l^{(0)}, l^{(1)}, \cdots, l^{(L-1)} \right) | l^{(k)} = d^{(k)} - E\left(d^{(k+1)} \right), \quad k = 0, \cdots, L-2, \; l^{(L-1)} = d^{(L-1)} \right\} \tag{8}$$

is calculated, which is sufficient for reconstruction of the DFD at the refinement layer decoder. The base layer decoder needs only to decode the truncated Laplacian pyramid $\boldsymbol{L}'$. The Laplacian pyramid is quantized with a layered quantization scheme. The coefficients are selected and quantized according to their amplitude in decreasing order. Due to the energy concentration, mainly coefficients from higher levels are coded first. The resulting symbol streams are coded using conditioning contexts and adaptive arithmetic coding [8]. Since the context choice depends only on the current or higher levels, scalability is retained. A detailed description of the coding scheme and the filters is given in Section V.

The open-loop approach, where the Laplacian pyramid levels are quantized independently, simplifies layered quantization, and allows for embedded coding. Furthermore, as is shown in (41) in Appendix A, the feedback of the quantization errors of the Laplacian pyramid within the DPCM loop does not degrade the performance since the current level $l_n^{(k)}$ is affected only by the quantization noise of the corresponding level of the previous frame $l_{n-1}^{(k)}$.

The Laplacian pyramid coding principle is adopted for coding of the motion vector fields. The main differences are that for both vector field components independent Laplacian pyramids are calculated, and the pyramids are coded losslessly. The vector field of the base layer is decomposed into Laplacian pyramids, and to obtain the complete pyramid, the motion vector differences of the refinement layer are added as level 0. Since the vector field components have finite precision, quantization is replaced by bit plane coding. This topic is elaborated in Section VI.

III. Motion Estimation and Compensation

The motion model used for this coder concept assumes that the motion of the 3-D objects can be described completely by displacements $v_n(\boldsymbol{x})$ on the image plane. Denoting the location of the pixels on the image grid by $\boldsymbol{x} = (x, y)^T$, frame g_n is related to g_{n-1} by

$$g_n(\boldsymbol{x} + v_n(\boldsymbol{x})) = g_{n-1}(\boldsymbol{x})$$

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

neglecting occlusions, uncovered background, and global illumination changes. The displacement vector field for frame g_n is the set of all displacement vectors v_n.

The aim of motion estimation in the context of coding is to find displacements $\hat{v}_n(\boldsymbol{x})$ with respect to the previously decoded frame $\tilde{g}_{n-1}$

$$\hat{v}_n = \mathrm{ME}(g_n, \tilde{g}_{n-1})$$

such that the prediction error

$$e_{\mathrm{DFD}} = \sum_{\boldsymbol{x}} \|g_n(\boldsymbol{x}) - \hat{g}_n(\boldsymbol{x})\| \tag{9}$$

becomes minimal according to a norm. The prediction signal $\hat{g}_n$ is obtained typically by displacing pixels of the previous frame, which may be weighted additionally (e.g., overlap block motion compensation) using a weighting function w_i

$$\hat{g}_n(\boldsymbol{x}) = \sum_i w_i(\boldsymbol{x}) \cdot \tilde{g}_{n-1}(\boldsymbol{x} - \hat{v}_n(\boldsymbol{x}_i)) = \mathrm{MC}(\tilde{g}_{n-1}, \hat{v}_n). \tag{10}$$

A. Optimal Motion Compensation for the Base Layer

For the moment, we assume that the original previous frame is used for "motion compensation." On the base layer, motion compensation is performed on low-pass filtered and downsampled frames, which limits the prediction performance even if the motion is known. One reason is that subsampling causes aliasing errors. Another one is that motion compensation and low-pass filtering cannot be interchanged in general

$$f_R\Big(\mathrm{MC}\Big(g_{n-1}^{(k)}, v^{(k)}\Big)\Big) \approx \mathrm{MC}\Big(f_R\Big(g_{n-1}^{(k)}\Big), v^{(k)}\Big) \tag{11}$$

because the low-pass filter $f_R(\cdot)$ weights and sums up spatially neighbored pixels. Thus, the filtered pixels are different if the original pixels are displaced prior to filtering [13]. An exception is the case of constant displacements $v(\boldsymbol{x}) = \text{const.}$, where all pixels undergo the same displacement. Low-pass filtering and motion compensation are then interchangeable, and equality holds in (11).

A deeper analysis reveals that motion can be interpreted as a shearing of the image spectrum into a hyperplane along the ω_t axis in the 3-D frequency space, and motion compensation reverts the shearing [14]. Therefore, low-pass filtering prior to motion compensation causes degradations since the hyperplane rather than the plane of the original image spectrum is multiplied by the filter transfer function. This interpretation also reveals why motion compensation prior to filtering results in more accurate predictions than compensation after filtering.

According to our motion model, where the moving scene content results in displacements of pixels in the image plane of the camera, the next frame is generated by displacing the pixels of the previous frame followed by filtering and subsampling. Therefore, for motion compensation of the base layer using a given motion vector field $v_n^{(0)}$, the low-pass filtered prediction from the highest available resolution level results in a higher prediction gain than calculating the prediction on the lower resolution level [13]

$$\hat{g}_{n,\mathrm{opt}}^{(k)} = R^k\Big(\mathrm{MC}\Big(g_{n-1}^{(0)}, v_n^{(0)}\Big)\Big), \qquad k = 0, 1. \tag{12}$$

This type of prediction is termed in the following *optimum prediction*. The superscript k denotes that the reduce function is applied k times. In principle, (12) holds also if the *reconstructed* previous frame is used for motion compensation. Hence, the Gaussian-type pyramid $\boldsymbol{D}_n$ formed by the displaced frame differences

$$d_n^{(k)} = g_n^{(k)} - \hat{g}_{n,\mathrm{opt}}^{(k)}, \qquad k = 1 \cdots L-1$$

has minimal energy. Furthermore, due to the linearity of the reduce operator

$$\begin{aligned} R(d_n^{(k)}) &= R\Big(g_n^{(k)} - \hat{g}_{n,\mathrm{opt}}^{(k)}\Big) \\ &= R\Big(g_n^{(k)}\Big) - R\Big(\hat{g}_{n,\mathrm{opt}}^{(k)}\Big) \\ &= g_n^{(k+1)} - \hat{g}_{n,\mathrm{opt}}^{(k+1)} \\ &= d_n^{(k+1)} \end{aligned}$$

holds for each level of $\boldsymbol{D}_n$. Thus, $\boldsymbol{D}_n$ is identical to the pyramid obtained by decomposing the prediction error of level $k = 0$. It is shown in [8] that such a pyramid can be coded efficiently.

B. Scalable Motion Compensation

This is the crucial part in a spatially scalable coding scheme since the prediction for the base layer needs to be calculated without considering the refinement level. The design aim with respect to (2) is to achieve equality of

$$R\Big(\mathrm{MC}\Big(g_{n-1}^{(0)}, v_n^{(0)}\Big)\Big) \approx \mathrm{MC}\Big(R\Big(g_{n-1}^{(0)}\Big), v_n^{(1)}\Big) \tag{13}$$

with $v_n^{(1)}$ being a suitably reduced version of $v_n^{(0)}$. Due to (11), an optimal prediction for the base layer $g_n^{(1)}$ cannot be obtained from $g_{n-1}^{(1)}$ in general. Therefore, the aim is to approximate $\hat{g}_{n,\mathrm{opt}}^{(1)}$ as closely as possible which, with respect to (12), is equivalent to approximating the refinement layer $g_{n-1}^{(0)}$ as accurately as possible. The idea is to interpolate the base layer frame $g_{n-1}^{(1)}$ such that the distance to the refinement layer $g_{n-1}^{(0)}$ is minimized in the least squares sense:

$$\Big|g_{n-1}^{(0)} - E\Big(g_{n-1}^{(1)}\Big)\Big|^2 \longrightarrow \min. \tag{14}$$

In real coders, only the *coded* previous frame $\tilde{g}_{n-1}^{(k)} = g_{n-1}^{(k)} + q'^{(k)}_{n-1}$ is available, where q' denotes the quantization error (Appendix A). However, (14) remains valid since the quantization error of level 0 is fixed

$$\tilde{g}_{n-1}^{(0)} - E\Big(\tilde{g}_{n-1}^{(1)}\Big) = g_{n-1}^{(0)} - E\Big(g_{n-1}^{(1)}\Big) + q_{n-1}^{(0)}.$$

Motion compensation is performed on the expanded frames, and the final prediction is obtained by reducing the compensated, expanded frame

$$\hat{g}_n^{(k)} = R\Big(\mathrm{MC}\Big(E\Big(\tilde{g}_{n-1}^{(k)}\Big), v_n^{(k)}\Big)\Big), \qquad k = 0, 1. \tag{15}$$

Since the equation can be interpreted for the base layer as half-pel motion compensation (assuming interger valued $v_n^{(0)}$), the same technique is also applicable for the refinement layer

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

$k = 0$. The vector fields $v_n^{(k)}$ are obtained by motion estimation on the expanded frames

$$v_n^{(k)} = \mathrm{ME}\Big(E\Big(g_n^{(k)}\Big), E\Big(\tilde{g}_{n-1}^{(k)}\Big)\Big), \qquad k = 0, 1. \tag{16}$$

Note that (15) depends on no particular method for motion estimation and compensation.

To achieve a good approximation of $\hat{g}_{n,\mathrm{opt}}^{(1)}$, the operator pair $E(\cdot), R(\cdot)$ needs to meet further constraints. The chain of operators must equal the identity operator

$$R\Big(E\Big(g_{n-1}^{(k)}\Big)\Big) = g_{n-1}^{(k)}.$$

This characteristic ensures that filtering effects do not distort the prediction. Furthermore, for the special case of global constant translational motion, the optimum prediction can be obtained.

Additionally, the filters must account for the fact that motion is a local property. Hence, the filters should be spatially localized or, in other words, the impulse response of the filters should have compact support.

The required properties are provided by centered third-order spline filters, a modified version of the cubic spline filters described in [15]. The class of spline filters allows for a tradeoff between spatial localization and aliasing since they include the Haar filter as a filter of order 0 as well as the sinc interpolator for infinite order. The distinctive feature of the proposed filters is the centering of the lower resolution grid with respect to the higher resolution grid. The advantage for scalable motion compensation is that the centered spline filters do not introduce a half-pel shift, and displacements remain at the same location in different levels. In Section IV, the filter design is linked to the Laplacian pyramids, and a detailed description is given in Appendix B.

C. Implementation Aspects

Since in a scalable video coding scheme the refinement level is not available to the base layer encoder, motion estimation is performed top down starting at the base layer. The method for motion estimation and compensation can be chosen freely in principle. However, an approach based on block matching is simple and robust. Another advantage of a block-oriented scheme is that low-pass filtering and motion compensation are interchangeable for $v_n^{(k)} = \mathrm{const}$. (11). Therefore, locally using a translational motion model is advisable.

To obtain a smooth vector field, a gain–cost criterion is evaluated as distance measure for each block j [16]

$$v_n(j) = \underset{v \in \mathcal{V}}{\operatorname{argmin}} \left\{ \log\Big(e_j^2(v)\Big) + \lambda \cdot \sum_{i \in \mathcal{N}_j} \|v - v_n(i)\| \right\} \qquad \lambda = \mathrm{const}. \tag{17}$$

with $\mathcal{N}_j$ denoting the set of neighbored blocks of block j. The calculation of the prediction error e_j for block j according to (9) employs overlapping blocks; hence, (17) includes overlap block motion estimation. Furthermore, motion compensation causes no blocking artifacts. The factor λ controls the smoothness by penalizing large vector differences. Due to the interaction with neighbored blocks, the vector field needs to be calculated iteratively. However, the computational load is still less than for full search block matching since the test vector set $\mathcal{V}$ is very small (18 candidates), and only about four iterations are sufficient. Since estimating motion at the expanded frames (16) can be interpreted as half-pel motion estimation, no additional half-pel motion estimation procedure like bilinear interpolation is needed.

At the refinement layer, the same gain–cost constrained block matching scheme applies. The vector field $v_n^{(1)}$ from the base layer is taken into consideration. Due to the constraint of (2), the predictions must be consistent across scales. Hence, the size of the blocks is increased by a factor of 2. A block size of 16×16 at level 0 corresponds to a block size of $16 \cdot 2^{-k} \times 16 \cdot 2^{-k}$ at level k. As a consequence, the vector field sizes of the scalable resolution levels are *equal*. The vector field $\hat{E}_v(v_n^{(1)})$ serves as an initial estimate for motion estimation, calculated by an appropriate scaling function $\hat{E}_v$. Since the number of vectors at both levels is the same, $\hat{E}_v$ scales only the amplitudes of the vector components by 2. Regarding $v_n^{(k)}$ as average motion, the motion vector field $v_n^{(k-1)}$ can be written as

$$v_n^{(k-1)} = \hat{E}_v\Big(v_n^{(k)}\Big) + \Delta v_n^{(k-1)} \tag{18}$$

where $\Delta v_n^{(k-1)}$ denotes the refinement. The gain–cost criterion constrains the motion estimate to smooth vector fields eliminating spurious vectors. Finally, a hierarchy of motion vector fields has been obtained, and needs to be coded (Section VI).

IV. Choice of Reduce and Expand Operators

The choice of the reduce operator $R(\cdot)$ and the expand operator $E(\cdot)$ is the key issue to achieve efficient multiresolution motion estimation and compensation, as well as efficient coding of the Laplacian pyramid of the displaced frame differences.

In the previous section, we compiled operator conditions for multiresolution motion compensation. First, the interpolation of the base layer should approximate the refinement layer in the least squares sense. Furthermore, the operator chain $E(\cdot)$ followed by $R(\cdot)$ should be the identity operator.

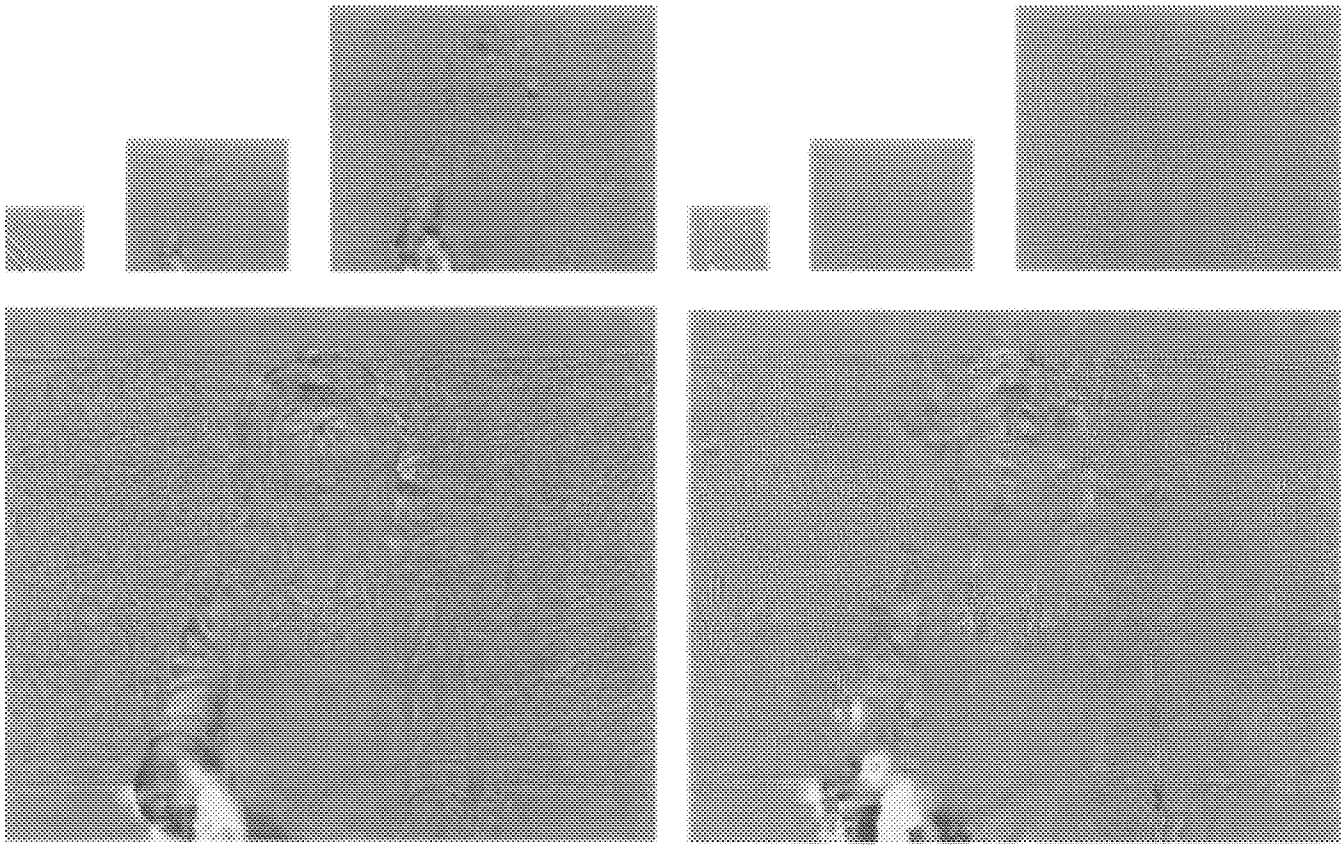
For DFD coding, the main objective for the filter design with respect to the Laplacian pyramid is to shift the signal energy into the higher pyramid levels. Concentration of the energy in only few pixels results in the desired coding gain. This is equivalent to the objective of minimizing the energy of the lower levels of the Laplacian pyramid. A useful criterion is that the approximation operator [$R(\cdot)$ followed by $E(\cdot)$] should be optimal in the least squares sense.

Comparing both sets of conditions, it turns out that they are identical for both tasks. Hence, one set of filters suffices. We chose spline filters, and found that centered third-order filters performed best. In Appendix B, the filters are described in detail, and it is proven that they provide the desired properties.

V. DFD Pyramid Coding

The two-layer scalable coding scheme under consideration outputs two displaced frame differences $d^{(0)}, d^{(1)}$ for each coded frame. The base layer DFD $d^{(1)}$ is decomposed into

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 5. Least squares Laplacian pyramid (LSLP) decomposition of DFD 11 of sequence *Silent*. For display, the amplitudes have been multiplied by two for the Gaussian and by four for the Laplacian pyramid. (a) Gaussian DFD pyramid $\boldsymbol{D}$. (b) Laplacian DFD pyramid $\boldsymbol{L}$.

a Laplacian pyramid. As explained in Section II, a Laplacian level $l^{(0)}$ is derived from the refinement layer DFD $d^{(0)}$, and this level builds together with the truncated pyramid $\boldsymbol{L}'$ the complete Laplacian pyramid of both DFD's.

This results in energy concentration into the higher (low-pass) levels of the pyramid, which consist of considerably fewer pixels than the original image. Usually, the energy is not equally distributed within the levels. This is due to the nature of motion compensation: some areas in the original sequence may contain complex motion where the motion compensation fails, while other areas contain no or only simple translational motion. The DFD energy will be the highest in areas where motion compensation failed. The severely limited bit budget for DFD coding in low bit-rate coders only allows encoding of selected areas of the DFD. Most DFD coding schemes encode the address information as overhead (addressing the areas, which are actually coded) separately from the quantized data. We use instead a layered coding method, where address information and data are encoded jointly in significance maps.

The coding gain which can be achieved with the Laplacian pyramid depends strongly on the employed filters. We have chosen a centered version of the cubic spline *least squares Laplacian pyramid* (LSLP) because this set of filters performed best of all investigated linear filters (see Appendix B).

In Fig. 5, the LSLP decomposition is shown for the frame difference between frame 11 and the motion-compensated prediction for this frame obtained from the reconstructed frame 6 of the image sequence "silent." The lowest level of the pyramid consists of 352 × 288 pixels (common interchange format, CIF). In Fig. 5(a), the Gaussian DFD pyramid $\boldsymbol{D}$ is given; in Fig. 5(b), the corresponding Laplacian pyramid $\boldsymbol{L}$ is given. The top three images in each subfigure constitute the truncated pyramids $\boldsymbol{D}'$ and $\boldsymbol{L}'$ (base layer DFD and Laplacian pyramid). The large images $d^{(0)}$ and $l^{(0)}$ belong to the refinement layers.

One can see that the base layer decomposition of $d^{(1)}$ into the truncated pyramid $\boldsymbol{L}'$ yields energy compaction into the higher pyramid levels. The design objective $R(d^{(0)}) \approx d^{(1)}$ does not hold with equality; therefore, $l^{(0)}$ contains more energy than the other levels of the Laplacian pyramid. Especially where strong motion appears (the raising hand), $d^{(0)}$ cannot be predicted well from $d^{(1)}$. On the other hand, $l^{(0)}$ contains much less energy than $d^{(0)}$, which indicates that predicting $d^{(0)}$ from $d^{(1)}$ is better than no prediction at all. Doing without prediction leads to the simulcast case (independent coding of base and refinement layer), which is used for comparison in Section VII.

A. Layered Quantization

Let $l^{(l)}(i,j)$ denote the pixel value at location (i,j) of level l of the pyramid $\boldsymbol{L}$. The range of the indexes is $i \in 0 \cdots I/2^l - 1$, $j \in 0 \cdots J/2^l - 1$, and $l \in 0 \cdots L-1$, where I and J are the dimensions of the original frame. The pyramid is scanned top down (starting with $l = L-1$ down to $l = 0$), resulting in a

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

sequence of pyramid values $x[i]$. The scan order inside each level is a simple line scan.

Scalability is achieved by progressive quantization in a sequence of up to N layers. Each layer can be described by the associated quantizer. It is convenient to differentiate between two sets of quantizers $\mathcal{Q}_0^D, \cdots, \mathcal{Q}_{N-1}^D$ and $\mathcal{Q}_1^S, \cdots, \mathcal{Q}_{N-1}^S$. The first set of quantizers belongs to the *dominant passes*, and the second to the *subordinate passes* of the coding algorithm. These passes alternate, starting with a dominant pass. Hence, during the coding process, the sequence $\mathcal{Q}_0^D$, $\mathcal{Q}_1^S$, $\mathcal{Q}_1^D$, $\mathcal{Q}_2^S$, $\mathcal{Q}_2^D, \cdots$ of quantizers is employed.

Each quantizer $\mathcal{Q}_n$ is defined by a set of disjoint quantization intervals $\mathcal{I}_{n,k_1}, \mathcal{I}_{n,k_2}, \cdots$ and the quantization function

$$\mathcal{Q}_n^D(x) = k, \qquad \text{for } x \in \mathcal{I}_{n,k}^D \tag{19}$$

$$\mathcal{Q}_n^S(x) = k, \qquad \text{for } x \in \mathcal{I}_{n,k}^S \tag{20}$$

which maps each input value x to an index k.

In order to allow efficient layered quantization, the sequence of quantization intervals must form a set of nested intervals.

The "dominant" intervals $\mathcal{I}_{n,k}^D$ are symmetric around zero, and are uniformly spaced with a dead zone twice as large as all of the other quantization intervals. From layer to layer, each quantization interval is halved. Thus, the intervals are specified with a single parameter Δ_0 as follows:

$$\mathcal{I}_{n,k}^D = \begin{cases} (-\Delta_n, \Delta_n), & \text{if } k = 0 \\ [k\Delta_n, (k+1)\Delta_n), & \text{if } k > 0 \\ ((k-1)\Delta_n, k\Delta_n], & \text{if } k < 0 \end{cases} \tag{21}$$

with

$$\Delta_n = \Delta_{n-1}/2, \qquad \text{for } n > 0. \tag{22}$$

The "subordinate" intervals $\mathcal{I}_{n,k}^S$ are refinements of the intervals $\mathcal{I}_{n-1,k}^D$. More precisely, each interval $\mathcal{I}_{n-1,k}^D$ contains two intervals $\mathcal{I}_{n,k}^S$ of equal width, except for $k = 0$, where $\mathcal{I}_{n-1,0}^D = \mathcal{I}_{n,0}^S$. Hence, the "subordinate" intervals are symmetric around zero, uniformly spaced with a dead zone four times larger than the other intervals

$$\mathcal{I}_{n,k}^S = \begin{cases} \mathcal{I}_{n-1,0}^D, & \text{if } k = 0 \\ \emptyset, & \text{if } k \in \{-1, 1\} \\ \mathcal{I}_{n,k}^D, & \text{if } k \notin \{-1, 0, 1\}. \end{cases} \tag{23}$$

Note that each quantization interval $\mathcal{Q}_n^D$ is contained in some quantization interval of $\mathcal{Q}_{n-1}^D$. Moreover, the intervals $\mathcal{I}_{n,0}^D$ contain the *three* intervals $\mathcal{I}_{n+1,-1}^D$, $\mathcal{I}_{n+1,0}^D$, $\mathcal{I}_{n+1,1}^D$, whereas all other intervals $\mathcal{I}_{n,k}^D$ (with $k \neq 0$) contain *two* intervals $\mathcal{I}_{n+1,2k}^D$, $\mathcal{I}_{n+1,2k+1}^D$. (Similar properties exist for the "subordinate" intervals since these are defined through (23) via the dominant intervals.) This is the only condition on the quantization intervals which must be imposed. However, specifying the intervals as in (21) is very convenient because the set of quantization functions is defined by Δ_0 and N alone.

B. Symbol Stream Generation

We define two sets of sequences $\sigma_n'[i]$ and $\delta_n'[i]$ as follows:

$$\sigma_n'[i] = \mathcal{Q}_n^D(x[i]) \tag{24}$$

$$\delta_n'[i] = \mathcal{Q}_n^S(x[i]) \pmod 2. \tag{25}$$

Since the quantization intervals form a set of nested intervals, all information necessary to recover the current quantization intervals is contained in these sequences. Moreover, if these sequences are ordered in the same way as the quantizers, i.e., $\sigma_0'[i]$, $\delta_1'[i]$, $\sigma_1'[i], \cdots$, a part of this sequence is still redundant. Particularly, 1) every $\delta_n'[i]$ for which $\sigma_{n-1}'[i] = 0$, and 2) every value $\sigma_n'[i] \notin \{-1, 0, 1\}$ can be predicted from previous parts of the alternating sequence $\sigma_0'[i]$, $\delta_1'[i]$, $\sigma_1'[i], \cdots$.

Thus, the variable length strings σ_0, δ_1, $\sigma_1, \cdots$ contain the same information as the strings $\sigma_0'[i]$, $\delta_1'[i]$, $\sigma_1'[i], \cdots$, if we denote by σ_n and δ_n the strings obtained by removal of the redundant entries of $\sigma_n'[i]$ and $\delta_n'[i]$, respectively.

Removal of the redundant entries works the same way as in Shapiro's EZW coder [6]. In the first dominant pass, all entries are scanned; thus, σ_0' equals σ_0. The first subordinate pass deals only with the samples which have become significant in the first dominant pass. In accordance with condition 1), those values of $\delta_1'[i]$, for which $\sigma_0'[i]$ has been zero (insignificant), are omitted in δ_1. Condition 2) means that all samples which have become significant during previous dominant passes are not regarded in any subsequent dominant pass.

The symbol stream consisting of the concatenation of the strings σ_0, $\delta_1, \cdots$ is encoded arithmetically, which is described in the next subsection.

C. Conditional Arithmetic Coding

The strings δ_n are binary, and show only very little correlation between the letters. Thus, they are encoded with an adaptive arithmetic coder without regarding any context.

The three-valued strings σ_n are encoded with an arithmetic coder conditioned on context [17], which is collected from previously reconstructed values of $\sigma_n'[i]$. For this task, a conditioning sequence $\kappa_n[i]$ is constructed from $\sigma_n'[i]$ by thresholding

$$\kappa_n[i] = \begin{cases} -1, & \text{if } \sigma_n'[i] < 0 \\ 0, & \text{if } \sigma_n'[i] = 0 \\ +1, & \text{if } \sigma_n'[i] > 0. \end{cases} \tag{26}$$

Denote by (j, k) the coordinates and by l the pyramid level of the pixel belonging to the index i. For convenience, we set $\kappa_n^l(j, k) = \kappa_n[i]$. Then the values of $\kappa_n^l(j-1, k)$ and $\kappa_n^l(j, k-1)$ are used to build a local conditioning context. Additionally, the significance information of $\kappa_n^{l+1}(j/2, k/2)$ (i.e., $|\kappa_n^{l+1}(j/2, k/2)|$) is used to utilize correlations of significance across scales. The implementation of the arithmetic coder follows the paper of Witten *et al.* [18].

D. Inverse Quantization at the Decoder

Until now, the pyramid encoder has been described. The decoder receives an arithmetically encoded bit stream, decodes it into a stream of symbols, and reconstructs from this symbol stream successively refined quantization intervals. For each coefficient, a value of the current interval must be chosen as the reconstruction value (inverse quantization). The reconstructed Laplacian levels are suitably expanded and summed up, giving the base and refinement layer DFD's $d^{(1)}$ and $d^{(0)}$.

There are several possibilities for designing the inverse quantization function. In Shapiro's original EZW coder, the

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

reconstruction values are chosen in the middle of the current interval. Since our pyramid decomposition is overcomplete, there is some redundancy in the unquantized pyramid. This means that not all possible combinations are consistent with the employed reduce and expand operators. We utilize this redundancy in the pyramid for quantization error reduction.

Theorem 1: Denote by $l_{Q'}^{(k)}$ the kth Laplacian level with independently quantized values. Then the quantized Laplacian levels

$$l_Q^{(k)} = l_{Q'}^{(k)} - E\Big(R\Big(l_{Q'}^{(k)}\Big)\Big), \qquad k = 1, \cdots, L-2 \tag{27}$$

are, in general, closer (in the mean-squared sense) to the unquantized levels $l^{(k)}$ than the levels $l_{Q'}^{(k)}$.

Proof: For least squares pyramids, the concatenation of reduce and expand operators yields the identity operator or an orthogonal projection operator P (depending on the order)

$$R(E(x)) = x$$

and

$$P(x) := E(R(x)) = P(P(x)), \qquad \forall x. \tag{28}$$

These properties are derived in Appendix B.

All "true" Laplacian levels (i.e., the levels $l^{(k)}$, $k = 1, \cdots, L-2$) have the form $l^{(k)} = d^{(k)} - E(R(d^{(k)})) = d^{(k)} - P(d^{(k)})$. Keeping this in mind, it is easy to see that the projection of an unquantized Laplacian level is zero

$$P(l) = E(R(l)) = 0. \tag{29}$$

Denote by $l_{Q'}$ a Laplacian level with independently quantized values. We now decompose these levels into a sum of the true Laplacian level and a quantization error

$$l_{Q'} = l + q. \tag{30}$$

Application of the projection operator P on both sides leads to

$$P(l_{Q'}) = P(q) \tag{31}$$

and

$$l_{Q'} - P(l_{Q'}) = l_{Q'} - P(q) = l + q - P(q). \tag{32}$$

Thus, application of (27) projects the quantization error q linearly into the null space of P. Hence, for the new quantization error $q - P(q)$, we have $\|q - P(q)\| \leq \|q\|$ with equality if and only if q belongs to the null space of P. ■

To utilize the above-mentioned redundancy of the pyramid decomposition, we first choose reconstruction values independently based on the quantization intervals known to the decoder. Then we project the quantization error into the vector space of admissible Laplacian levels by applying (27).

VI. Displacement Vector Field Coding

An efficient and elegant coding method is derived by adopting the pyramid approach for coding of displaced frame differences. Specific aspects are that vector fields need to be coded lossless, and each vector consists of two components. The concept of conditioning contexts is employed because it is much more flexible and even more efficient than zerotree coding [10]. The flexibility is especially important for the scalable codec.

At first, coding of the base layer motion vector field $v_n^{(1)}$ is described. Similar to the coding approach for the prediction error, both components v_x and v_y of the motion vector field $v_n^{(1)}$ are decomposed separately into a Laplacian pyramid $\boldsymbol{L}_x$ and $\boldsymbol{L}_y$ of $L_v - 1$ levels, respectively. For simplicity, only one component is mentioned in the following, where no ambiguity can occur. The levels of the pyramid are obtained by

$$\begin{aligned} l_x^{(k)} &= v_x^{(k)} - E_v\Big(v_x^{(k+1)}\Big), \qquad k = 1, \cdots, L_v - 2 \\ l_x^{(L_v-1)} &= v_x^{(L_v-1)} \end{aligned} \tag{33}$$

where

$$v_x^{(k)} = R_v\Big(v_x^{(k-1)}\Big), \qquad k = 2, \cdots, L_v - 1 \tag{34}$$

denotes the levels of the intermediate Gaussian pyramid. The reduce and expand operators $R_v(\cdot)$, $E_v(\cdot)$ need to be designed carefully to obtain optimal compression. On the one hand, a high energy concentration into the higher pyramid levels should be achieved; on the other hand, one should be able to code the remaining residuals at lower levels efficiently. The two important properties of the coding technique of significance maps are that zero coefficients can be coded efficiently, and that one large residual coefficient is more expensive to code than two small coefficients.

The operators $R_v(\cdot)$ and $E_v(\cdot)$ have a 2×2 support to obtain a quadtree-like pyramid structure. As expand operator $E_v(\cdot)$, simple repetition is employed. The nonlinear operator $R_v(\cdot)$ used for reduction termed *closest couple* provides the best performance in our experiments. To achieve that at least one coefficient of the Laplacian pyramid becomes zero, this operator outputs one element of a block of 2×2 vector components. The selection criterion maximizes the number of small coefficients by first searching for the coefficient pair which has the smallest difference. The coefficient of the pair which has the smallest difference to the remaining two values is taken as output. Since half-pel motion compensation is used, the vector field components are scaled by a factor of 2 and treated as integers. Hence, the coefficients of the Laplacian pyramid also have integer precision.

The vector fields must be coded lossless, and the integer precision allows for simple bit plane coding, which could also be interpreted as a specialized version of layered "quantization." Instead of calculating a threshold Δ as for DFD coding, the position of the most significant bit is calculated. Halving of the threshold is equivalent to switching to the next lower bit plane. Significant coefficients are the coefficients with a bit set in the currently selected bit plane. Therefore, similar to DFD coding, a significance sequence σ_n is obtained for each bit plane, which is coded employing conditioning contexts followed by adaptive arithmetic coding [18]. The formulation of the different contexts is based on the conditioning sequence $\kappa_n[i]$.

Due to the quadtree structure, one class of contexts depends only on conditioning symbols $\kappa_n[i]$ within a 2×2 block. To keep the context causal, unavailable conditioning symbols $\kappa_n[i]$ are taken from the previous layer $\kappa_{n-1}[i]$. From the

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

3^3 possible states, the seven most frequently occurring states (experimentally determined) are used. Since, in the most significant bit plane, only a few coefficients become significant while most coefficients become significant later, the set of contexts is switched after scanning the second bit plane. In the case that all coefficients for a block are insignificant, the corresponding conditioning symbol of the next higher level is taken into account.

Finally, a different set of contexts is defined for the top level of the pyramid, which has a low-pass characteristic and no block structure, in contrast to the band-pass characteristic of the lower levels. This set is the same as used for DFD coding, except that no information on higher levels is available.

Regarding the scalable video codec, a hierarchy of the vector fields is generated (Section III-C). The vector fields do not form a pyramid since $v^{(0)}$ and $v^{(1)}$ have the same number of vectors. For calculating the refinement level $l_x^{(0)}$ of the Laplacian pyramid in (33), (18) is directly applicable

$$l_x^{(0)} = v_x^{(1)} - 2 \cdot \left(v_x^{(1)} \right) \tag{35}$$

replacing the expand function E_v by the scaling function $\hat{E}_v$. Since the decoder knows the motion vector field resolution anyway, the refinement vector field can be reconstructed from the Laplacian "pyramid." For coding, the concept of conditioning contexts is still well suited because the symbols denoting the bit plane (quantization interval) can be calculated independently of the higher resolution levels. Moreover, the conditioning contexts for entropy coding of the symbol stream are designed such that they depend only on elements from the same level.

VII. Simulation Results

First, the suitability of the motion estimation and compensation scheme (15) in the context of the proposed scalable coder is demonstrated. Frames 6, 11, and 16 of the sequence *foreman*[4] have been coded with a nonscalable codec as well as with a scalable codec. The base layer prediction error has been decomposed into a Gaussian pyramid of four levels. Correspondingly, in the nonscalable coder, the DFD has been decomposed into five levels. For both codecs, the highest spatial resolution is CIF, hence, the base layer (level 1) has QCIF resolution. For motion estimation and compensation, blocks of 32 × 32 are matched on the refinement level; thus, the block size on level 1 is 16 × 16. Both coders run at the same bit rate of 96 kbit/s at 5 frames/s (fps), and since the coder control assigns a constant bit budget to each frame, the coding results can be compared. The first frame (1) has been coded with 48 kbits.

In the nonscalable coder, the prediction for the "base level" is obtained by reducing the prediction of the "refinement level," according to (12). This is equivalent to a reduction of the reconstructed refinement layer to obtain the reconstructed base layer frame. The prediction gains measured in decibels for the nonscalable and the scalable codec in Table I are comparable.

[4]The sequences are obtained from the original CCIR sequences using least squares cubic spline filters.

TABLE I
PSNR in Decibels of the Resolution Levels After Motion Compensation

Frame	Level	Nonscalable	Scalable
6	0	26.9	27.1
	1	27.7	27.9
11	0	28.7	28.5
	1	30.2	29.8
16	0	29.8	29.2
	1	31.6	30.9

The diagrams in Fig. 6 show the overall coding performance for the sequence *silent*, coded at 96 kbit/s and 5 fps with both a scalable coder as well as a nonscalable coder. The base layer DFD is decomposed into four levels. For motion compensation, blocks of 8 × 8 on the base layer and 16 × 16 on the refinement layer have been used. The pyramid for coding the base layer vector field has three levels.

The PSNR of the base layer compared to the coding performance of a nonscalable coder (QCIF) running at the same average bit rate as the scalable coder (56 kbit/s) is given in Fig. 6(a). Except for the first frames, the performance is similar, as expected. For the refinement layer, the PSNR values are shown in Fig. 6(b). The comparison with the nonscalable coder running at the same total bit rate of 96 kbit/s indicates an upper bound. Furthermore, the results are compared with the simulcast case. The average bit rate of 40 kbit/s devoted to the refinement layer in the scalable coder is used to code the CIF sequence with the nonscalable coder independently.

On hand of the following diagrams, the coder is analyzed in more detail. Regarding the rate allocation shown in Fig. 6(c), the bit rate per frame is absolutely constant, while a varying part is allocated by the base layer. The difference is the rate available for the refinement layer. With only about 40 kbit/s, the additional refinement layers can be transmitted at a sufficient quality. As an example, the coded refinement layer image 61 is shown in Fig. 7.

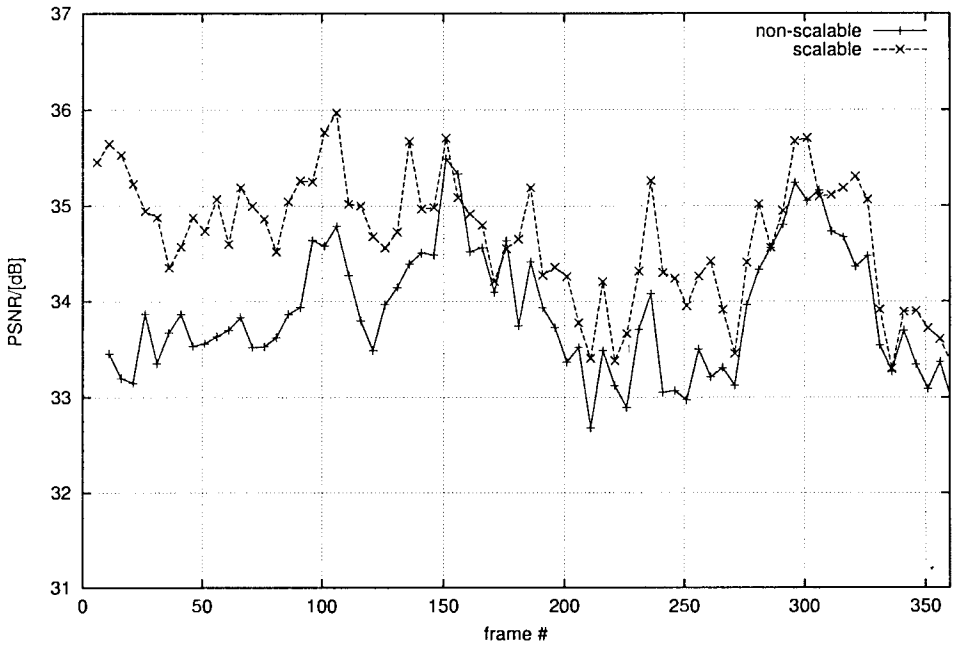
The diagram in Fig. 8(a) shows solely the motion compensation performance of the base layer compared to a nonscalable QCIF coder. Both coders run in a closed loop at a fixed bit rate including update coding, but only the PSNR after motion compensation before update coding is shown. As expected, the gain is similar. In Fig. 8(b), the performance of the refinement layer compared to a nonscalable CIF coder running at 96 kbit/s is given. Although the curves differ due to different bit allocations, the overall motion compensation gain is comparable.

As is shown in Fig. 9, coding the CIF sequence *container* at 48 kbit/s and 5 fps with the scalable coding scheme also results in a gain compared to the simulcast case, where the refinement level coder runs at 25 kbit/s.

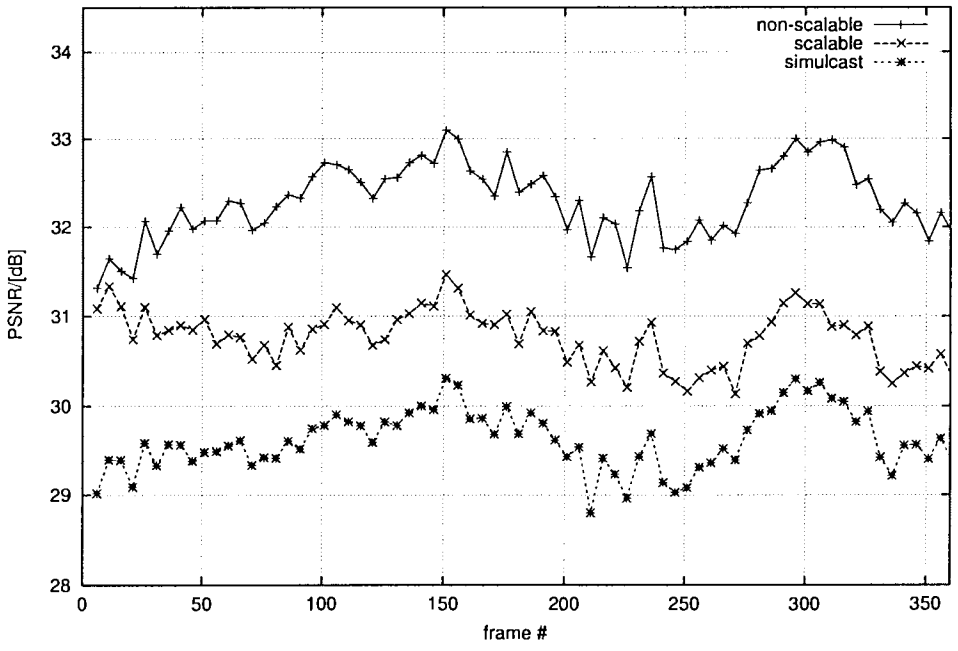
VIII. Characteristics of the Proposed Coding Method

The proposed scheme for a spatially scalable coder relies on a predictive motion-compensated approach. In contrast to spatiotemporal coding methods which utilize correlations of more than two frames (e.g., 3-D subband coding or motion estimation with more than one frame), such coders do not necessarily introduce a delay of more than one frame. The update information (the coded DFD) is calculated on a frame-by-

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.



(a)



(b)

Fig. 6. Coding performance of the scalable coder for the sequence *silent* at 5 fps. (a) PSNR of the base layer compared to the nonscalable coder running at the same bit rate (56 kbit/s). (b) PSNR of the refinement layer compared to the nonscalable coder running at the same bit rate (96 kbit/s) and the simulcast case (nonscalable CIF coder at 40 kbit/s).

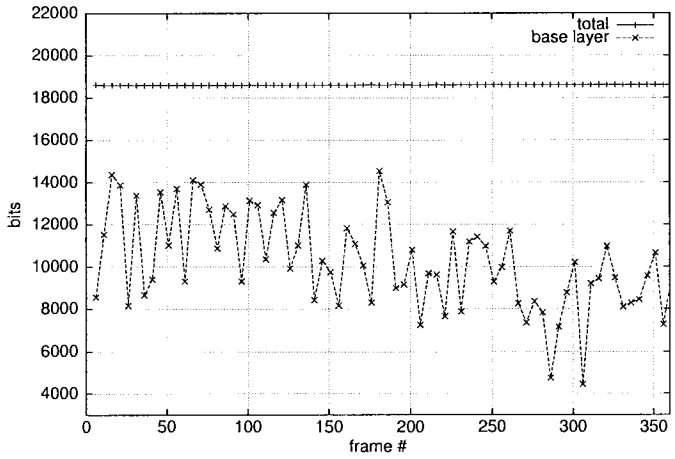
frame basis, thus retaining this low-delay property. Therefore, the proposed coder is especially suited for communications applications.

One key feature is the decomposition of the displaced frame differences into a Laplacian pyramid. In connection with overlap block motion compensation, the coder produces no blocking artifacts. The artifacts are similar to the artifacts of subband coders and are dependent on the used filters.

There is no need for a special start-up procedure since rough (low-pass) approximations of large areas are possible with a constrained bit rate. On the other hand, if most parts of the DFD are close to zero, the available bit rate can be spent in small regions, thus allowing an update of fine details. This property simplifies coder control considerably after a scene cut. Furthermore, there is no need for an explicit inter/intraswitch because the Laplacian pyramid is suited for static image statistics as well.

If only still images are transmitted (or if there is only slight motion present), the reconstructed image at the receiver will converge to a perfect reconstruction. This means that still images are automatically progressively coded (see Appendixes A–B).

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.



(c)

Fig. 6. (*Continued.*) Coding performance of the scalable coder for the sequence *silent* at 5 fps. (c) Rate allocation for the base layer and for the complete frame.

(a) (b)

Fig. 7. Frame 61 of test sequence *silent* coded with the scalable coder (CIF, 96 kbit/s, 5 fps). (a) Coded refinement layer of frame 61 at 40 kbit/s. (b) Close up.

The DFD coder outputs an embedded bit stream, which eases coder control and adds flexibility. The sequence coder control may truncate the bit stream at *any* point (usually when the available bit budget is exhausted, or the buffer exceeds a given threshold). Thus, the coder has the ability to code frames at a *constant* bit rate, eliminating the need for a buffer, and thus reducing the delay. Since there is no separate encoding of address information and quantized data, the algorithm works efficiently in a wide range of bit rates.

IX. Discussion and Summary

A new spatially scalable predictive video coding scheme is described in this paper. The basic feature is a predictive coding approach for the base layer as well as for the refinement layer, such that the displaced frame differences can be decomposed into a *single* Laplacian pyramid.

One major aspect of the paper is the specific design of motion estimation and compensation. Motion compensation is performed such that the displaced frame difference of the base layer becomes similar to the reduced prediction error of the refinement layer. Therefore, motion is estimated and compensated on interpolated frames approximating the higher resolution level in the least square sense. As filters, centered cubic spline filters are chosen. The relation of the motion-compensated predictions on the base and the refinement layer determines the complete coder performance. Further research is directed to the improvement of the robustness when the predictions differ significantly.

Motion is estimated in a hierarchical fashion, taking into account the motion vector field of the next lower resolution level. Since the difference between levels represents an additional refinement information, a similar approach as for coding of the displaced frame differences is designed to encode the vector fields efficiently using a Laplacian pyramid coding approach employing conditioning contexts.

The second main aspect is the design of an improved Laplacian pyramid decomposition of the base layer. It turns out that centered cubic spline filters are among the best linear

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

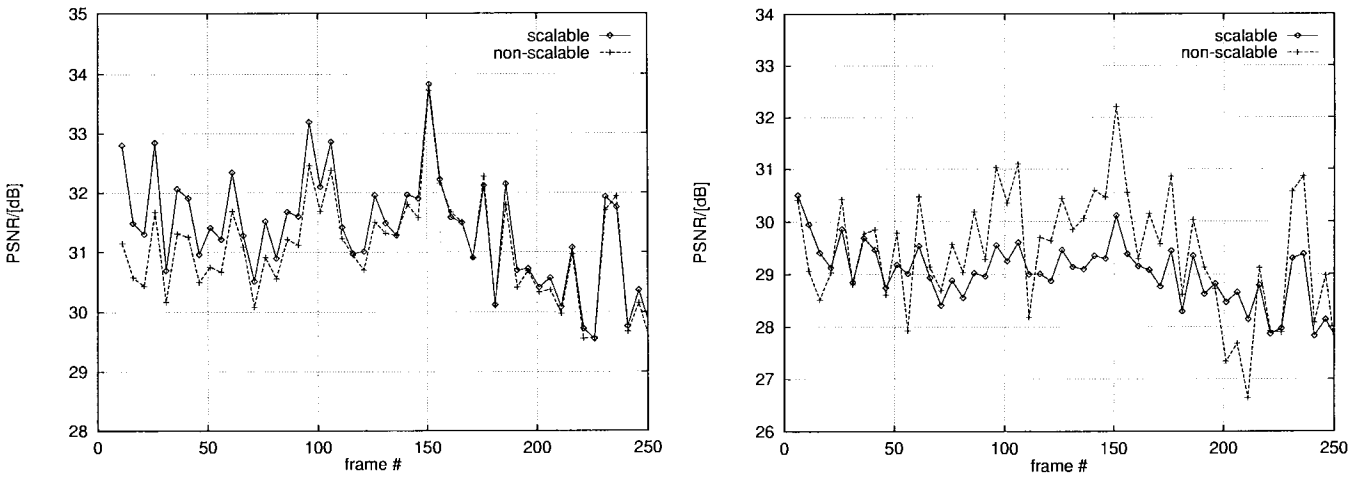


Fig. 8. Motion compensation performance. (a) Motion compensation gain of the base layer compared to a nonscalable QCIF coder. (b) Motion compensation gain of the refinement layer compared to a nonscalable CIF coder.

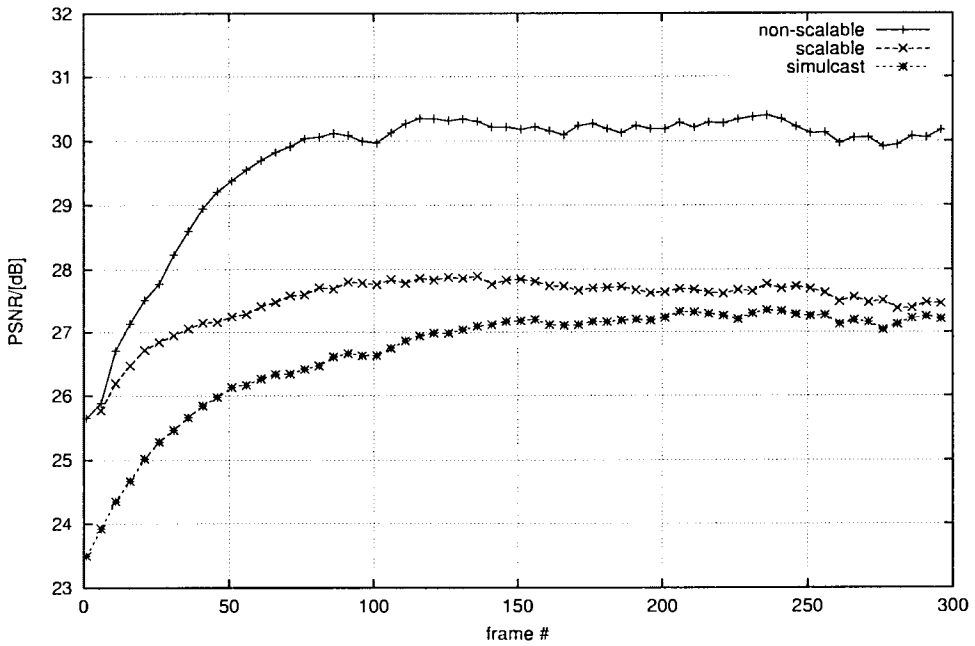


Fig. 9. Coding performance of the scalable coder for the sequence *container* at 5 fps.

filters fulfilling the desired design objectives. Interestingly, the *same* filters meet the constraints for scalable motion compensation as well as for efficient pyramid decomposition.

Furthermore, a rate-distortion efficient embedded DFD coding method using conditioning contexts for coding is developed, which allows for progressive transmission and simplifies coder control significantly. Simulation results are provided for the complete hybrid video coder.

APPENDIX A
ANALYSIS OF THE QUANTIZATION ERRORS

A. Open-Loop Quantization

The effects of quantizing the Laplacian DFD pyramid $\boldsymbol{L}$ in an open loop are analyzed in the following. In an open-loop approach the quantization error $q^{(k)}$ at level k is independent of quantization errors at other levels $q^{(i)}, i \neq k$

$$\tilde{l}^{(k)} = l^{(k)} + q^{(k)} \tag{36}$$

The coded displaced frame difference $\tilde{d}^{(k)}$ at level k is then given by

$$\begin{aligned}\tilde{d}^{(k)} &= \tilde{l}^{(k)} + E\left(\tilde{d}^{(k+1)}\right) \\ &= l^{(k)} + q^{(k)} + \sum_{i=k+1}^{L-1} E^{i-k}\left(l^{(i)}\right) + \sum_{i=k+1}^{L-1} E^{i-k}\left(q^{(i)}\right).\end{aligned}$$

With the convention $E^0(x) = x$, this leads to

$$\tilde{d}^{(k)} = d^{(k)} + \sum_{i=k}^{L-1} E^{i-k}\left(q^{(i)}\right). \tag{37}$$

Thus, the quantization error $q'^{(k)}$ of the displaced frame

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

difference at level k is given by

$$q'^{(k)} := \tilde{d}^{(k)} - d^{(k)} = \sum_{i=k}^{L-1} E^{i-k}\left(q^{(i)}\right). \tag{38}$$

Based on this relation, the quantization error for the coded frame is given by

$$\begin{aligned}\tilde{g}^{(k)} &= \hat{g}^{(k)} + \tilde{d}^{(k)} \\ &= g^{(k)} + q'^{(k)}.\end{aligned} \tag{39}$$

Therefore, the difference between the coded refinement image (level 0) and the coded and expanded base layer (level 1) is affected only by the quantization error of the refinement layer

$$\begin{aligned}\tilde{g}^{(0)} - E\left(\tilde{g}^{(1)}\right) &= g^{(0)} + q'^{(0)} - E\left(g^{(1)} + q'^{(1)}\right) \\ &= g^{(0)} - E\left(g^{(1)}\right) + q^{(0)}.\end{aligned} \tag{40}$$

B. Quantization Error Feedback

Furthermore, the quantization error feedback due to the predictive coding structure can be analyzed. Therefore, we assume no motion compensation (all displacement vectors are zero). Hence, the prediction is just the previously coded frame $\hat{g}_n^{(k)} = \tilde{g}_{n-1}^{(k)}$. The refinement level of the Laplacian pyramid to be coded

$$\begin{aligned}l_n^{(0)} &= d_n^{(0)} - E\left(d_n^{(1)}\right) \\ &= g_n^{(0)} - \hat{g}_n^{(0)} - \left(E\left(g_n^{(1)}\right) - E\left(\hat{g}_n^{(1)}\right)\right) \\ &= g_n^{(0)} - \tilde{g}_{n-1}^{(0)} - \left(E\left(g_n^{(1)}\right) - E\left(\tilde{g}_{n-1}^{(1)}\right)\right)\end{aligned}$$

depends on the reconstructed previous base and refinement level. Using (40) reveals

$$l_n^{(0)} = g_n^{(0)} - E\left(g_n^{(1)}\right) - \left(g_{n-1}^{(0)} - E\left(g_{n-1}^{(1)}\right)\right) - q_{n-1}^{(0)} \tag{41}$$

that $l_n^{(0)}$ is affected only by the quantization error $q_{n-1}^{(0)}$ of this level. No shift of quantization errors across scales occurs. Therefore, coding of still images and unchanged regions between subsequent frames converges to lossless coding.

Appendix B
Derivation of the Reduce and Expand Operators

In this Appendix, reduce and expand operators are derived, which are based on spline approximations. The derivation follows in its main aspects [15], [9], and is given in terms of discrete one-dimensional operators. For construction of the pyramids used for motion estimation and for DFD coding, these operators are extended to the 2-D case by separable application along the rows and columns of the frames.

The main difference between [9] and our approach is the centering of the lower resolution grid with respect to the high-resolution grid. This yields several advantages compared to the classical approach. First, the centering leads to a quadtree-like topology of the pyramid. Thus, each pixel on a coarse pyramid level belongs to four corresponding pixels of the next finer level, which is advantageous for the propagation of block-based motion vector fields across the differing resolutions. Second, consistent boundary conditions are obtained by mirrored extension of the finite signals, while the uncentered pyramid requires periodic extension which degrades the coding performance at signal support boundaries.

A. Discrete Least Squares Approximation

We consider the construction of a dyadic pyramid in a purely discrete framework. The basic operation is the approximation from a fine space S_1 onto a coarse space S_2 at twice the scale. For convenience, we focus first on sequences of infinite length. Implications of the finite support of the signals are discussed in the following subsection. Thus, we set $S_1 = l_2$ (the space of squared summable sequences), and consider the coarser subspace $S_2 \subset l_2$ generated from the even integer translates of a generating sequence h:

$$\begin{aligned}S_2 &= \operatorname{span}\{h[k-2l]\}_{l\in\mathbb{Z}} \\ &= \left\{\tilde{s}_1 | \tilde{s}_1[k] = \sum_{k\in\mathbb{Z}} s_2[l]h[k-2l], s_2 \in l_2\right\}.\end{aligned} \tag{42}$$

We may think of $s_2[k]$ as being samples of some pyramid level. Then $\tilde{s}_1[k]$ are the samples of the corresponding interpolated (expanded) level. We therefore define the (one-dimensional) expand operator E as

$$E : l_2 \to l_2; \quad s_2 \mapsto E(s_2) = [s_2]_{\uparrow 2} * h \tag{43}$$

and S_2 becomes simply the image of E

$$S_2 = \{\tilde{s}_1 | \tilde{s}_1 = E(s_2), s_2 \in l_2\}. \tag{44}$$

Note that the generating sequence h has a twofold meaning here: 1) it generates the Riesz basis $\{h[k-2l]\}_{l\in\mathbb{Z}}$ of S_2, and hence serves as a scaling function, and 2) it equals the impulse response of the synthesis filter, thus defining the expand operator.

The reduce operator R consists of an antialiasing filter (or analysis filter) with impulse response $\overset{\circ}{h}$ followed by a subsampler:

$$R : l_2 \to l_2; \quad s_1 \mapsto s_2 = R(s_1) = \left[s_1 * \overset{\circ}{h}\right]_{\downarrow 2}. \tag{45}$$

The concatenation of R and E approximates any sequence $s_1 \in l_2$ with a sequence $\tilde{s}_1 = E(R(s_1)) \in S_2$. A prefilter $\overset{\circ}{h}$ is called *optimal* in the least squares sense with respect to a synthesis filter h when the energy $\Sigma_{k\in\mathbb{Z}}(s_1[k] - \tilde{s}_1[k])^2$ of the approximation error becomes minimal.

Theorem 2: The optimal prefilter with respect to a synthesis filter with z transform $H(z)$ is given by

$$\overset{\circ}{H}(z) = \frac{2H(z^{-1})}{H(z)H(z^{-1}) + H(-z)H(-z^{-1})}. \tag{46}$$

Proof: The least squares approximation is achieved when the error $s_1[k] - \tilde{s}_1[k]$ is orthogonal to S_2 or, equivalently

$$\langle h[k-2l], s_1[k]\rangle = \left\langle h[k-2l], \sum_{n\in\mathbb{Z}} s_2[n]h[k-2n]\right\rangle, \quad \forall l \in \mathbb{Z}.$$

With $h^T[k] = h[-k]$, we can rewrite the inner products as convolutions

$$[h^T * s_1]_{\downarrow 2} = [h^T * h]_{\downarrow 2} * s_2.$$

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

Application of the inverse of $[h^T * h]_{\downarrow 2}$ on both sides (which exists because we require h constituting a Riesz basis) yields

$$\begin{aligned} s_2 &= ([h^T * h]_{\downarrow 2})^{-1} * [h^T * s_1]_{\downarrow 2} \\ &= \left[\left[([h^T * h]_{\downarrow 2})^{-1}\right]_{\uparrow 2} * h^T * s_1\right]_{\downarrow 2} \\ &= \left[\overset{\circ}{h} * s_1\right]_{\downarrow 2}. \end{aligned}$$

Thus, the optimal prefilter has the impulse response $\overset{\circ}{h} = [([h^T * h]_{\downarrow 2})^{-1}]_{\uparrow 2} * h^T$, whose z transform is given by (46). ■

Theorem 3: Least squares pyramids have the property that reduction of an expanded signal leaves the signal unchanged.

Proof: By definition, the reduce operator R of a least squares pyramid is chosen such that the energy of the approximation error is minimized. This being true, the concatenated operator

$$P : l_2 \to S_2, \quad s \mapsto \tilde{s} = E(R(s))$$

must be an orthogonal projector from l_2 into S_2. This implies $P(P(s)) = P(s)$. For the reduce and expand operators, this means that $E(R(E(R(s)))) = E(R(s))$, and

$$R(E(s)) = s, \qquad \forall s \in l_2 \tag{47}$$

follows immediately. ■

This property is crucial for the scalable coding scheme described in this paper.

B. Centered Spline Approximation

Until now, we have not specified the space S_2 and the filter h. We confine ourselves to spline pyramids mainly for to reasons. First, splines have excellent approximation properties [19], [20]. Second, the spline framework allows for a progressive transition between the piecewise constant and the band-limited model. The spline of degree zero leads to the Haar wavelet (best time localization), and the cardinal spline filters converge to the sinc interpolator for high degrees [21]. So it can be expected that a reasonable tradeoff between time localization and aliasing can be found inside the class of spline filters.

Usually, the generating kernel h is symmetric, and the coarser grid points are positioned on the even integers. We propose instead to shift the sampling grid of the approximating function such that the samples "sit" centered between the samples of the high-resolution grid. Using the formalism developed in [15], it is not difficult to derive the z transform $H_\Delta(z)$ of the cardinal spline interpolator corresponding to a shift Δ

$$H_{\Delta(z)} = \frac{B^n_{2,\Delta}(z)}{B^n_{1,0}(z)} \tag{48}$$

where the sampled B-spline filter kernels $B^n_{m,\Delta}(z)$ are defined as

$$B^n_{m,\Delta}(z) = \sum_{k\in\mathbb{Z}} \beta^n\left(\frac{k-\Delta}{m}\right) z^{-k} \tag{49}$$

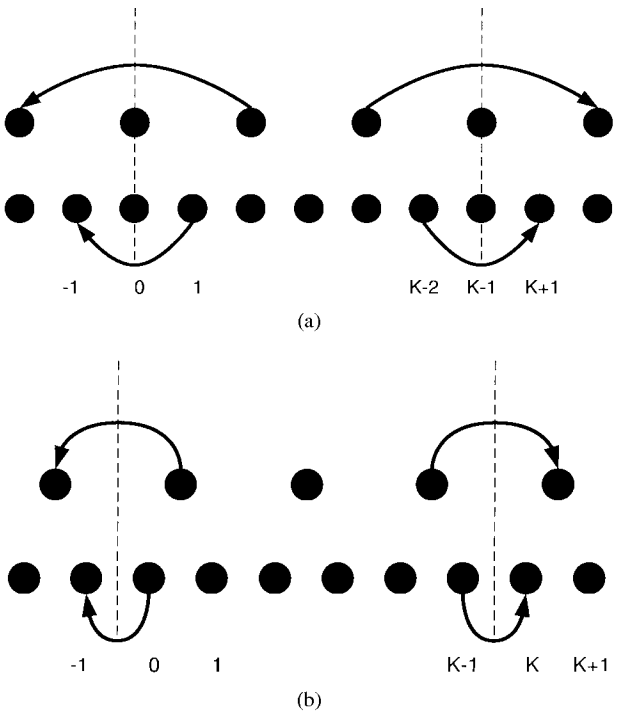


Fig. 10. Odd and even mirroring of a discrete signal. (a) Odd mirroring in case of a traditional pyramid. (b) Even mirroring in case of a centered pyramid.

and $\beta^n(x)$ is the symmetrical B spline of degree n, defined by

$$\beta^n(x) = \beta^0(x) * \beta^{n-1}(x)$$

with

$$\beta^0(x) = \mathrm{rect}(x) = \begin{cases} 1, & \text{for } |x| \le 1/2 \\ 0, & \text{for } |x| > 1/2. \end{cases}$$

For the shift parameter $\Delta = 0$, the pyramid equals the one derived in [9], and corresponds to the traditional formulation of the discrete wavelet transform. For $\Delta = 1$, the samples of the coarse grid sit on the odd integers, and for $\Delta = 1/2$, we end up with the *centered* pyramid which we use for the compression algorithm. In Fig. 10, the positioning of the sampling grids is depicted for the traditional ($\Delta = 0$) and the centered ($\Delta = 1/2$) case. The centered pyramid allows us to assign each sample on the fine grid the closest sample on the coarse grid, whereas this is possible in the traditional pyramid only for the even samples.

Until now, only infinite signals have been considered. In the case of finite signals, we have to impose suited boundary conditions. In signal processing, such boundary conditions are usually formulated in terms of signal extensions across the boundaries. It is easy to show that in the case of the uncentered spline approximation, consistent boundary conditions are given by periodic repetition of the signals. However, this introduces unnatural discontinuities at the boundaries, which degrade the performance of most signal processing applications. Therefore, a symmetric extension by mirroring is preferred in many applications, as depicted in Fig. 11 for continuous signals.

If the length of the signal K is even, a proper symmetric extension is impossible as long as we stay with the uncentered grid. On the other hand, if we use a centered grid as proposed

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

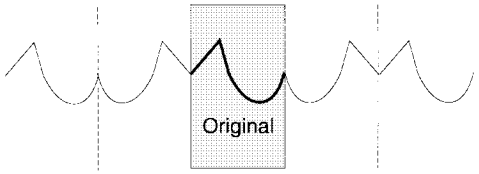


Fig. 11. Symmetric boundary extension of a continuous signal with finite support.

in this paper, we can use the "even" mirroring

$$s[-k-1] = s[k], \qquad k = 0 \cdots K-1$$
$$s[k] = s[2K-1-k], \qquad k = K \cdots 2K-1$$

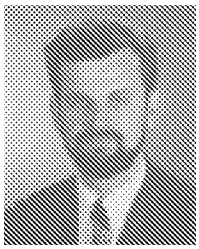
without distorting property (47).

ACKNOWLEDGMENT

The authors would like to thank Prof. Dr.-Ing. H. D. Lüke, Aachen University of Technology, Germany, for his support of the underlying research, and also the anonymous referees for their valuable comments.

REFERENCES

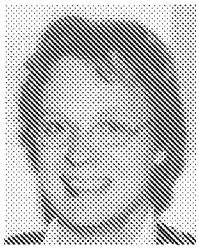
[1] T. Hanamura, W. Kameyama, and H. Tominaga, "Hierarchical coding scheme of video signals with scalability and compatibility," *Signal Processing: Image Commun.*, vol. 5, pp. 159–184, Feb. 1993.
[2] B. Girod, U. Horn, and B. Belzer, "Scalable video coding with multiscale motion compensation and unequal error protection," in *Proc. Symp. Multimedia Communication Video Coding*, New York, NY, Oct. 1995.
[3] P.-Y. Cheng, J. Li, and C.-C. Jay Kuo, "Multiscale video compression using wavelet transform and motion compensation," in *Proc. IEEE Int. Conf. Image Processing*, vol. I, Washington DC, 1995, pp. 606–609.
[4] T. Naveen and J. W. Woods, "Motion compensated multiresolution transmission of high definition video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 29–41, Feb. 1994.
[5] K. Tsunashima, J. B. Stampleman, and V. M. Bove, "A scalable motion-compensated subband image coder," *IEEE Trans. Commun.*, vol. 42, pp. 1894–1901, Feb./Mar./Apr. 1994.
[6] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
[7] D. Taubmann and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Processing*, vol. 3, pp. 572–588, Sept. 1994.
[8] F. Müller, K. Illgner, and B. Menser, "Embedded Laplacian pyramid image coding using conditional arithmetic coding," in *Proc. IEEE Int. Conf. Image Processing, ICIP'96*, vol. I, Lausanne, Switzerland, Sept. 1996, pp. 221–224.
[9] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part II—Efficient design and applications," *IEEE Trans. Signal Processing*, vol. 41, pp. 834–848, May 1993.
[10] K. Illgner and F. Müller, "Hierarchical coding of motion vector fields," in *Proc. IEEE Int. Conf. Image Processing, ICIP'95*, vol. I, Washington DC, Oct. 1995, pp. 566–569.
[11] ITU-T International Telecommunication Union, *Draft ITU-T Recommendation H.263 (Video Coding for Low Bitrate Communication)*, KPN Research, The Netherlands, Jan. 1995.
[12] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532–540, Apr. 1983.
[13] O. Werner, "Drift analysis and drift reduction for multiresolution video coding," *Signal Processing: Image Commun.*, vol. 8, pp. 387–409, May 1996.
[14] B. Girod, "Motion compensation: Visual aspects, accuracy, and fundamental limits," in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk, Eds. Boston: Kluwer Academic, 1993, pp. 125–152.
[15] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I—Theory," *IEEE Trans. Signal Processing*, vol. 41, pp. 821–833, May 1993.
[16] K. Illgner and F. Müller, "Motion estimation using overlapped block motion compensation and Gibbs-modeled vector fields," in *Proc. 9th Workshop Image and Multidimensional Signal Processing (IMDSP'96)*, Belize City, Belize, Mar. 1996, pp. 126–127.
[17] J. Rissanen and G. G. Langdon, "Universal modeling and coding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 12–23, Jan. 1981.
[18] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, June 1987.
[19] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.
[20] M. Unser, "Approximation power of biorthogonal wavelet expansions," *IEEE Trans. Signal Processing*, vol. 44, pp. 519–527, Mar. 1996.
[21] A. Aldroubi, M. Unser, and M. Eden, "Cardinal spline filters: Stability and convergence to the ideal sinc interpolator," *Signal Processing*, vol. 28, pp. 127–138, 1992.



Klaus Illgner received the Dipl. degree (M.S.) in electrical engineering from Aachen University of Technology (RWTH), Germany, in 1991.

Since then, he has been with the Institut für Elektrische Nachrichtentechnik, RWTH Aachen, as a Scientific Assistant. His research interests cover all aspects of low-bit-rate video coding and video communications, where his main focus at present is on motion compensation and multiresolution video.

Mr. Illgner is a member of the IEEE Signal Processing Society, the IEEE Communications Society, and the German Electrical Engineering Association (VDE).



Frank Müller received the Dipl. degree (M.S.) in electrical engineering from Aachen University of Technology (RWTH), Germany, in 1991.

Since then, he has been with the Institut für Elektrische Nachrichtentechnik, RWTH Aachen, as a Scientific Assistant. His current interests are hierarchical and multiresolution approaches to image and video compression, and source coding theory.

Mr. Müller is a member of the IEEE Signal Processing Society, the IEEE Information Theory Society, the German Electrical Engineering Association (VDE), and the German Computer Society (Gesellschaft für Informatik (GI)).

Authorized licensed use limited to: Rennae Johnson. Downloaded on August 10,2010 at 17:06:52 UTC from IEEE Xplore. Restrictions apply.

INTERNATIONAL STANDARD

ISO/IEC 13818-1

Second edition
2000-12-01

Information technology — Generic coding of moving pictures and associated audio information: Systems

Technologies de l'information — Codage générique des images animées et du son associé: Systèmes

Reference number
ISO/IEC 13818-1:2000(E)

© ISO/IEC 2000

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 2000

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

 © ISO/IEC 2000 – All rights reserved

Contents

Page

SECTION 1 – GENERAL 1
1.1 Scope 1
1.2 Normative references 1
1.2.1 Identical Recommendations | International Standards 1
1.2.2 Paired Recommendations | International Standards equivalent in technical content 2
1.2.3 Additional references 2
SECTION 2 – TECHNICAL ELEMENTS 2
2.1 Definitions 2
2.2 Symbols and abbreviations 5
2.2.1 Arithmetic operators 5
2.2.2 Logical operators 6
2.2.3 Relational operators 6
2.2.4 Bitwise operators 6
2.2.5 Assignment 6
2.2.6 Mnemonics 6
2.2.7 Constants 7
2.3 Method of describing bit stream syntax 7
2.4 Transport Stream bitstream requirements 8
2.4.1 Transport Stream coding structure and parameters 8
2.4.2 Transport Stream system target decoder 9
2.4.2.1 System clock frequency 10
2.4.2.2 Input to the Transport Stream system target decoder 11
2.4.2.3 Buffering 12
2.4.2.4 Decoding 17
2.4.2.5 Presentation 17
2.4.2.6 Buffer management 17
2.4.2.7 T-STD extensions for carriage of ISO/IEC 14496 data 18
2.4.3 Specification of the Transport Stream syntax and semantics 18
2.4.3.1 Transport Stream 18
2.4.3.2 Transport Stream packet layer 18
2.4.3.3 Semantic definition of fields in Transport Stream packet layer 19
2.4.3.4 Adaptation field 20
2.4.3.5 Semantic definition of fields in adaptation field 21
2.4.3.6 PES packet 31
2.4.3.7 Semantic definition of fields in PES packet 31
2.4.3.8 Carriage of Program Streams and ISO/IEC 11172-1 Systems streams in the Transport Stream 40
2.4.4 Program specific information 41
2.4.4.1 Pointer 43
2.4.4.2 Semantics definition of fields in pointer syntax 43
2.4.4.3 Program association Table 43
2.4.4.4 Table_id assignments 44
2.4.4.5 Semantic definition of fields in program association section 44
2.4.4.6 Conditional access Table 45
2.4.4.7 Semantic definition of fields in conditional access section 45
2.4.4.8 Program Map Table 46
2.4.4.9 Semantic definition of fields in Transport Stream program map section 46
2.4.4.10 Syntax of the Private section 47
2.4.4.11 Semantic definition of fields in private section 48
2.4.4.12 Syntax of the Transport Stream section 49
2.4.4.13 Semantic definition of fields in the Transport Stream section 50

© ISO/IEC 2000 – All rights reserved

Page

2.5 Program Stream bitstream requirements 50
2.5.1 Program Stream coding structure and parameters 50
2.5.2 Program Stream system target decoder 51
2.5.2.1 System clock frequency 52
2.5.2.2 Input to the Program Stream system target decoder 52
2.5.2.3 Buffering 53
2.5.2.4 PES streams 54
2.5.2.5 Decoding and presentation 54
2.5.2.6 P-STD extensions for carriage of ISO/IEC 14496 data 54
2.5.3 Specification of the Program Stream syntax and semantics 54
2.5.3.1 Program Stream 54
2.5.3.2 Semantic definition of fields in Program Stream 55
2.5.3.3 Pack layer of Program Stream 55
2.5.3.4 Semantic definition of fields in program stream pack 56
2.5.3.5 System header 56
2.5.3.6 Semantic definition of fields in system header 56
2.5.3.7 Packet layer of Program Stream 58
2.5.4 Program Stream map 59
2.5.4.1 Syntax of Program Stream map 59
2.5.4.2 Semantic definition of fields in Program Stream map 59
2.5.5 Program Stream directory 60
2.5.5.1 Syntax of Program Stream directory packet 60
2.5.5.2 Semantic definition of fields in Program Stream directory 60
2.6 Program and program element descriptors 62
2.6.1 Semantic definition of fields in program and program element descriptors 62
2.6.2 Video stream descriptor 62
2.6.3 Semantic definitions of fields in video stream descriptor 64
2.6.4 Audio stream descriptor 65
2.6.5 Semantic definition of fields in audio stream descriptor 65
2.6.6 Hierarchy descriptor 65
2.6.7 Semantic definition of fields in hierarchy descriptor 65
2.6.8 Registration descriptor 66
2.6.9 Semantic definition of fields in registration descriptor 66
2.6.10 Data stream alignment descriptor 67
2.6.11 Semantics of fields in data stream alignment descriptor 67
2.6.12 Target background grid descriptor 67
2.6.13 Semantics of fields in target background grid descriptor 68
2.6.14 Video window descriptor 68
2.6.15 Semantic definition of fields in video window descriptor 69
2.6.16 Conditional access descriptor 69
2.6.17 Semantic definition of fields in conditional access descriptor 70
2.6.18 ISO 639 language descriptor 70
2.6.19 Semantic definition of fields in ISO 639 language descriptor 70
2.6.20 System clock descriptor 70
2.6.21 Semantic definition of fields in system clock descriptor 71
2.6.22 Multiplex buffer utilization descriptor 71
2.6.23 Semantic definition of fields in multiplex buffer utilization descriptor 71
2.6.24 Copyright descriptor 72
2.6.25 Semantic definition of fields in copyright descriptor 72
2.6.26 Maximum bitrate descriptor 72
2.6.27 Semantic definition of fields in maximum bitrate descriptor 72
2.6.28 Private data indicator descriptor 73
2.6.29 Semantic definition of fields in Private data indicator descriptor 73
2.6.30 Smoothing buffer descriptor 73
2.6.31 Semantic definition of fields in smoothing buffer descriptor 74
2.6.32 STD descriptor 74
2.6.33 Semantic definition of fields in STD descriptor 74
2.6.34 IBP descriptor 74
2.6.35 Semantic definition of fields in IBP descriptor 74

 © ISO/IEC 2000 – All rights reserved

Page

2.6.36 MPEG-4 video descriptor ... 75
2.6.37 Semantic definition of fields in MPEG-4 video descriptor ... 75
2.6.38 MPEG-4 audio descriptor ... 75
2.6.39 Semantic definition of fields in MPEG-4 audio descriptor ... 75
2.6.40 IOD descriptor ... 75
2.6.41 Semantic definition of fields in IOD descriptor ... 77
2.6.42 SL descriptor ... 77
2.6.43 Semantic definition of fields in SL descriptor ... 77
2.6.44 FMC descriptor ... 77
2.6.45 Semantic definition of fields in FMC descriptor ... 78
2.6.46 External_ES_ID descriptor ... 78
2.6.47 Semantic definition of fields in External_ES_ID descriptor ... 78
2.6.48 Muxcode descriptor ... 78
2.6.49 Semantics of fields in Muxcode descriptor ... 79
2.6.50 FmxBufferSize descriptor ... 79
2.6.51 Semantics of fields in FmxBufferSize descriptor ... 79
2.6.52 MultiplexBuffer descriptor ... 79
2.6.53 Semantics of fields in MultiplexBuffer descriptor ... 80
2.7 Restrictions on the multiplexed stream semantics ... 80
2.7.1 Frequency of coding the system clock reference ... 80
2.7.2 Frequency of coding the program clock reference ... 80
2.7.3 Frequency of coding the elementary stream clock reference ... 81
2.7.4 Frequency of presentation timestamp coding ... 81
2.7.5 Conditional coding of timestamps ... 81
2.7.6 Timing constraints for scalable coding ... 81
2.7.7 Frequency of coding P-STD_buffer_size in PES packet headers ... 82
2.7.8 Coding of system header in the Program Stream ... 82
2.7.9 Constrained system parameter Program Stream ... 82
2.7.10 Transport Stream ... 83
2.8 Compatibility with ISO/IEC 11172 ... 84
2.9 Registration of copyright identifiers ... 84
2.9.1 General ... 84
2.9.2 Implementation of a Registration Authority (RA) ... 84
2.10 Registration of private data format ... 85
2.10.1 General ... 85
2.10.2 Implementation of a Registration Authority (RA) ... 85
2.11 Carriage of ISO/IEC 14496 data ... 85
2.11.1 Introduction ... 85
2.11.2 Carriage of individual ISO/IEC 14496-2 and 14496-3 Elementary Streams in PES packets ... 85
2.11.2.1 Introduction ... 85
2.11.2.2 STD extensions for individual ISO/IEC 14496 elementary streams ... 86
2.11.3 Carriage of audiovisual ISO/IEC 14496-1 scenes and associated ISO/IEC 14496 streams ... 87
2.11.3.1 Introduction ... 87
2.11.3.2 Assignment of ES_ID values ... 87
2.11.3.3 Timing of ISO/IEC 14496 scenes and associated streams ... 88
2.11.3.4 Delivery timing of SL-packetized streams ... 89
2.11.3.5 Delivery timing of FlexMux streams ... 89
2.11.3.6 Carriage of SL-packetized streams in PES packets ... 89
2.11.3.7 Carriage of FlexMux streams in PES packets ... 89
2.11.3.8 Carriage of SL packets and FlexMux packets in sections ... 90
2.11.3.9 T-STD extensions ... 91
2.11.3.10 Carriage within a Transport Stream ... 93
2.11.3.11 P-STD Model for 14496 content ... 94
2.11.3.12 Carriage within a Program Stream ... 96

Annex A – CRC Decoder Model ... 97
A.0 CRC decoder model ... 97

© ISO/IEC 2000 – All rights reserved

Page

Annex B – Digital Storage Medium Command and Control (DSM-CC) 98
B.0 Introduction 98
B.0.1 Purpose 98
B.0.2 Future applications 98
B.0.3 Benefits 98
B.0.4 Basic functions 99
B.0.4.1 Stream selection 99
B.0.4.2 Retrieval 99
B.0.4.3 Storage 99
B.1 General elements 99
B.1.1 Scope 99
B.1.2 Overview of the DSM-CC application 99
B.1.3 The transmission of DSM-CC commands and acknowledgements 100
B.2 Technical elements 101
B.2.1 Definitions 101
B.2.2 Specification of DSM-CC syntax 101
B.2.3 Semantics of fields in specification of DSM-CC syntax 102
B.2.4 Control layer 102
Constraints on setting flags in DSM-CC control 102
B.2.5 Semantics of fields in control layer 103
B.2.6 Acknowledgement layer 104
Constraints on setting flags in DSM-CC control 104
B.2.7 Semantics of fields in Acknowledgement layer 105
B.2.8 Time code 106
Constraints on time code 106
B.2.9 Semantics of fields in time code 106

Annex C – Program Specific Information 107
C.0 Explanation of Program Specific Information in Transport Streams 107
C.1 Introduction 107
C.2 Functional Mechanism 107
C.3 The Mapping of Sections into Transport Stream Packets 108
C.4 Repetition Rates and Random Access 108
C.5 What is a Program? 108
C.6 Allocation of program_number 109
C.7 Usage of PSI in a Typical System 109
C.8 The Relationships of PSI Structures 110
C.8.1 Program Association Table 110
C.8.2 Program Map Table 110
C.8.3 Conditional Access Table 110
C.8.4 Network Information Table 111
C.8.5 Private_section() 111
C.8.6 Descriptors 112
C.9 Bandwidth Utilization and Signal Acquisition Time 112

Annex D – Systems Timing Model and Application Implications of this Recommendation | International Standard 115
D.0 Introduction 115
D.0.1 Timing Model 115
D.0.2 Audio and Video Presentation Synchronization 116
D.0.3 System Time Clock recovery in the decoder 118
D.0.4 SCR and PCR Jitter 120
D.0.5 Clock Recovery in the Presence of Network Jitter 121
D.0.6 System clock used for chroma sub-carrier generation 121
D.0.7 Component video and audio reconstruction 122
D.0.8 Frame Slipping 122
D.0.9 Smoothing of network jitter 123

 © ISO/IEC 2000 – All rights reserved

Page

Annex E – Data Transmission Applications 124
E.0 General considerations 124
E.1 Suggestion 124
Annex F – Graphics of Syntax for this Recommendation | International Standard 125
F.0 Introduction 125
F.0.1 Transport Stream syntax 125
F.0.2 PES packet 126
F.0.3 Program Association Section 127
F.0.4 CA section 127
F.0.5 TS program map section 128
F.0.6 Private section 128
F.0.7 Program Stream 129
F.0.8 Program Stream map 130
Annex G – General Information 131
G.0 General Information 131
G.0.1 Sync Byte Emulation 131
G.0.2 Skipped picture status and decoding process 131
G.0.3 Selection of PID Values 131
G.0.4 PES start_code emulation 131
Annex H – Private Data 132
H.0 Private Data 132
Annex I – Systems conformance and real-time interface 133
I.0 Systems conformance and real-time interface 133
Annex J – Interfacing Jitter-Inducing Networks to MPEG-2 Decoders 134
J.0 Introduction 134
J.1 Network compliance models 134
J.2 Network specification for jitter smoothing 135
J.3 Example decoder implementations 136
J.3.1 Network adapter followed by an MPEG-2 decoder 136
J.3.2 Integrated decoder 136
Annex K – Splicing Transport Streams 137
K.0 Introduction 137
K.1 The different types of splicing point 137
K.1.1 Ordinary splicing points 137
K.1.2 Seamless splicing points 137
K.2 Decoder behaviour on splices 138
K.2.1 On non-seamless splices 138
K.2.2 On seamless splices 138
K.2.3 Buffer Overflow 138
Annex L – Registration procedure 140
L.1 Procedure for the request of a Registered Identifier (RID) 140
L.2 Responsibilities of the Registration Authority 140
L.2.1 Contact information of the Registration Authority 140
L.3 Responsibilities of parties requesting an RID 140
L.4 Appeal procedure for denied applications 141
Annex M – Registration application form 142
M.1 Contact information of organization requesting a Registered Identifier (RID) 142
M.2 Statement of an intention to apply the assigned RID 142
M.3 Date of intended implementation of the RID 142
M.4 Authorized representative 142
M.5 For official use only of the Registration Authority 142
Annex N 143

© ISO/IEC 2000 – All rights reserved

Page

Annex O – Registration procedure 144
O.1 Procedure for the request of an RID 144
O.2 Responsibilities of the Registration Authority 144
O.3 Contact information for the Registration Authority 144
O.4 Responsibilities of parties requesting an RID 144
O.5 Appeal procedure for denied applications 145
Annex P – Registration application form 146
P.1 Contact information of organization requesting an RID 146
P.2 Request for a specific RID 146
P.3 Short description of RID that is in use and date system that was implemented 146
P.4 Statement of an intention to apply the assigned RID 146
P.5 Date of intended implementation of the RID 146
P.6 Authorized representative 146
P.7 For official use of the Registration Authority 146
Annex Q – T-STD and P-STD buffer models for ISO/IEC 13818-7 ADTS 147
Q.1 Introduction 147
Q.2 Leak Rate from Transport Buffer 147
Q.3 Buffer Size 147
Q.3.1 TBS_n: same as other audio 148
Q.3.2 BS_{mux}: different from other audio 148
Q.3.3 BS_{dec}: different from other audio 148
Q.3.4 BS_{oh}: different from other audio 148
Q.4 Conclusion 148
Annex R – Carriage of ISO/IEC 14496 scenes in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 150
R.1 Content access procedure for ISO/IEC 14496 program components within a Program Stream 150
R.2 Content access procedure for ISO/IEC 14496 program components within a Transport Stream 150
Annex S – Patents 153

 © ISO/IEC 2000 – All rights reserved

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 13818 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 13818-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*, in collaboration with ITU-T. The identical text is published as ITU-T Rec. H.222.0.

This second edition cancels and replaces the first edition (ISO/IEC 3818-1:1996), which has been technically revised.

ISO/IEC 13818 consists of the following parts, under the general title *Information technology — Generic coding of moving pictures and associated audio information*:

— *Part 1: Systems*

— *Part 2: Video*

— *Part 3: Audio*

— *Part 4: Conformance testing*

— *Part 5: Software simulation*

— *Part 6: Extensions for DSM-CC*

— *Part 7: Advanced Audio Coding (ACC)*

— *Part 9: Extension for real time interface for systems decoders*

— *Part 10: Conformance extensions for Digital Storage Media Command and Control (DSM-CC)*

Annex A forms a normative part of this part of ISO/IEC 13818. Annexes B to S are for information only.

© ISO/IEC 2000 – All rights reserved

Introduction

The systems part of this Recommendation | International Standard addresses the combining of one or more elementary streams of video and audio, as well as other data, into single or multiple streams which are suitable for storage or transmission. Systems coding follows the syntactical and semantic rules imposed by this Specification and provides information to enable synchronized decoding of decoder buffers over a wide range of retrieval or receipt conditions.

System coding shall be specified in two forms: the **Transport Stream** and the **Program Stream**. Each is optimized for a different set of applications. Both the Transport Stream and Program Stream defined in this Recommendation | International Standard provide coding syntax which is necessary and sufficient to synchronize the decoding and presentation of the video and audio information, while ensuring that data buffers in the decoders do not overflow or underflow. Information is coded in the syntax using time stamps concerning the decoding and presentation of coded audio and visual data and time stamps concerning the delivery of the data stream itself. Both stream definitions are packet-oriented multiplexes.

The basic multiplexing approach for single video and audio elementary streams is illustrated in Figure Intro. 1. The video and audio data is encoded as described in ITU-T Rec. H.262 | ISO/IEC 13818-2 and ISO/IEC 13818-3. The resulting compressed elementary streams are packetized to produce **PES packets**. Information needed to use PES packets independently of either Transport Streams or Program Streams may be added when PES packets are formed. This information is not needed and need not be added when PES packets are further combined with system level information to form **Transport Streams** or **Program Streams**. This systems standard covers those processes to the right of the vertical dashed line.

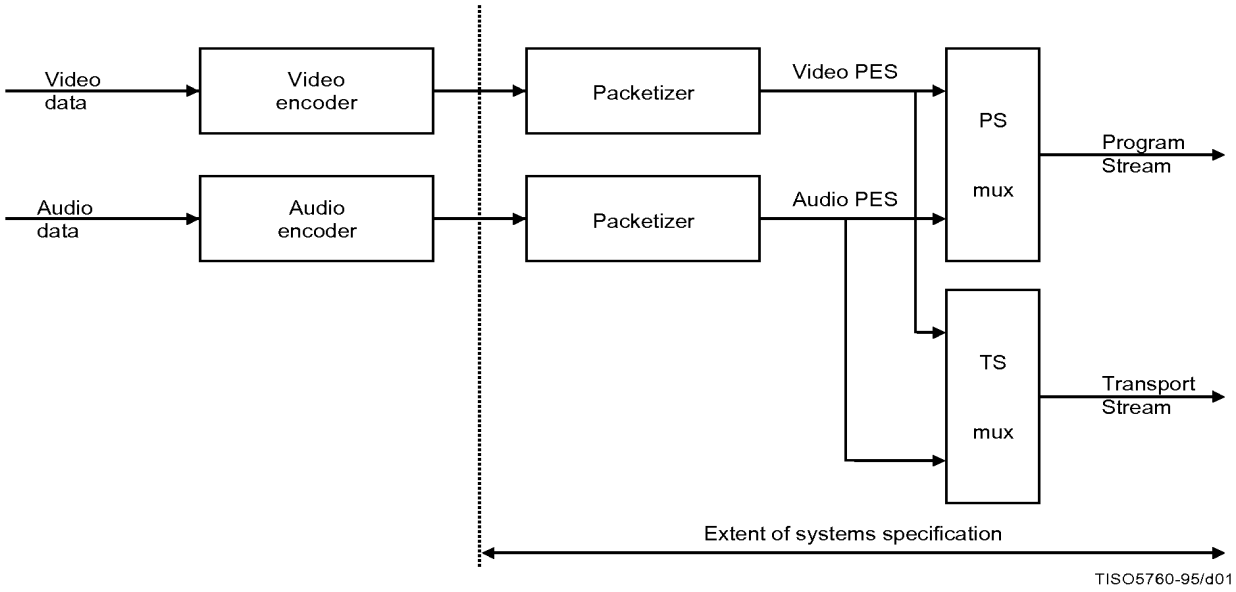


Figure Intro. 1 – Simplified overview of the scope of this Recommendation | International Standard

The **Program Stream** is analogous and similar to ISO/IEC 11172 Systems layer. It results from combining one or more streams of PES packets, which have a common time base, into a single stream.

For applications that require the elementary streams which comprise a single program to be in separate streams which are not multiplexed, the elementary streams can also be encoded as separate Program Streams, one per elementary stream, with a common time base. In this case the values encoded in the SCR fields of the various streams shall be consistent.

Like the single Program Stream, all elementary streams can be decoded with synchronization.

© ISO/IEC 2000 – All rights reserved

The Program Stream is designed for use in relatively error-free environments and is suitable for applications which may involve software processing of system information such as interactive multi-media applications. Program Stream packets may be of variable and relatively great length.

The **Transport Stream** combines one or more programs with one or more independent time bases into a single stream. PES packets made up of elementary streams that form a program share a common timebase. The Transport Stream is designed for use in environments where errors are likely, such as storage or transmission in lossy or noisy media. Transport Stream packets are 188 bytes in length.

Program and Transport Streams are designed for different applications and their definitions do not strictly follow a layered model. It is possible and reasonable to convert from one to the other; however, one is not a subset or superset of the other. In particular, extracting the contents of a program from a Transport Stream and creating a valid Program Stream is possible and is accomplished through the common interchange format of PES packets, but not all of the fields needed in a Program Stream are contained within the Transport Stream; some must be derived. The Transport Stream may be used to span a range of layers in a layered model, and is designed for efficiency and ease of implementation in high bandwidth applications.

The scope of syntactical and semantic rules set forth in the systems specification differ: the syntactical rules apply to systems layer coding only, and do not extend to the compression layer coding of the video and audio specifications; by contrast, the semantic rules apply to the combined stream in its entirety.

The systems specification does not specify the architecture or implementation of encoders or decoders, nor those of multiplexors or demultiplexors. However, bit stream properties do impose functional and performance requirements on encoders, decoders, multiplexors and demultiplexors. For instance, encoders must meet minimum clock tolerance requirements. Notwithstanding this and other requirements, a considerable degree of freedom exists in the design and implementation of encoders, decoders, multiplexors, and demultiplexors.

Intro. 1 Transport Stream

The Transport Stream is a stream definition which is tailored for communicating or storing one or more programs of coded data according to ITU-T Rec. H.262 | ISO/IEC 13818-2 and ISO/IEC 13818-3 and other data in environments in which significant errors may occur. Such errors may be manifested as bit value errors or loss of packets.

Transport Streams may be either fixed or variable rate. In either case the constituent elementary streams may either be fixed or variable rate. The syntax and semantic constraints on the stream are identical in each of these cases. The Transport Stream rate is defined by the values and locations of Program Clock Reference (PCR) fields, which in general are separate PCR fields for each program.

There are some difficulties with constructing and delivering a Transport Stream containing multiple programs with independent time bases such that the overall bit rate is variable. Refer to 2.4.2.2.

The Transport Stream may be constructed by any method that results in a valid stream. It is possible to construct Transport Streams containing one or more programs from elementary coded data streams, from Program Streams, or from other Transport Streams which may themselves contain one or more programs.

The Transport Stream is designed in such a way that several operations on a Transport Stream are possible with minimum effort. Among these are:

1) Retrieve the coded data from one program within the Transport Stream, decode it and present the decoded results as shown in Figure Intro. 2.

2) Extract the Transport Stream packets from one program within the Transport Stream and produce as output a different Transport Stream with only that one program as shown in Figure Intro. 3.

3) Extract the Transport Stream packets of one or more programs from one or more Transport Streams and produce as output a different Transport Stream (not illustrated).

4) Extract the contents of one program from the Transport Stream and produce as output a Program Stream containing that one program as shown in Figure Intro. 4.

5) Take a Program Stream, convert it into a Transport Stream to carry it over a lossy environment, and then recover a valid, and in certain cases, identical Program Stream.

© ISO/IEC 2000 – All rights reserved

Figure Intro. 2 and Figure Intro. 3 illustrate prototypical demultiplexing and decoding systems which take as input a Transport Stream. Figure Intro. 2 illustrates the first case, where a Transport Stream is directly demultiplexed and decoded. Transport Streams are constructed in two layers:

– a system layer; and

– a compression layer.

The input stream to the Transport Stream decoder has a system layer wrapped about a compression layer. Input streams to the Video and Audio decoders have only the compression layer.

Operations performed by the prototypical decoder which accepts Transport Streams either apply to the entire Transport Stream ("multiplex-wide operations"), or to individual elementary streams ("stream-specific operations"). The Transport Stream system layer is divided into two sub-layers, one for multiplex-wide operations (the Transport Stream packet layer), and one for stream-specific operations (the PES packet layer).

A prototypical decoder for Transport Streams, including audio and video, is also depicted in Figure Intro. 2 to illustrate the function of a decoder. The architecture is not unique – some system decoder functions, such as decoder timing control, might equally well be distributed among elementary stream decoders and the channel specific decoder – but this figure is useful for discussion. Likewise, indication of errors detected by the channel specific decoder to the individual audio and video decoders may be performed in various ways and such communication paths are not shown in the diagram. The prototypical decoder design does not imply any normative requirement for the design of a Transport Stream decoder. Indeed non-audio/video data is also allowed, but not shown.

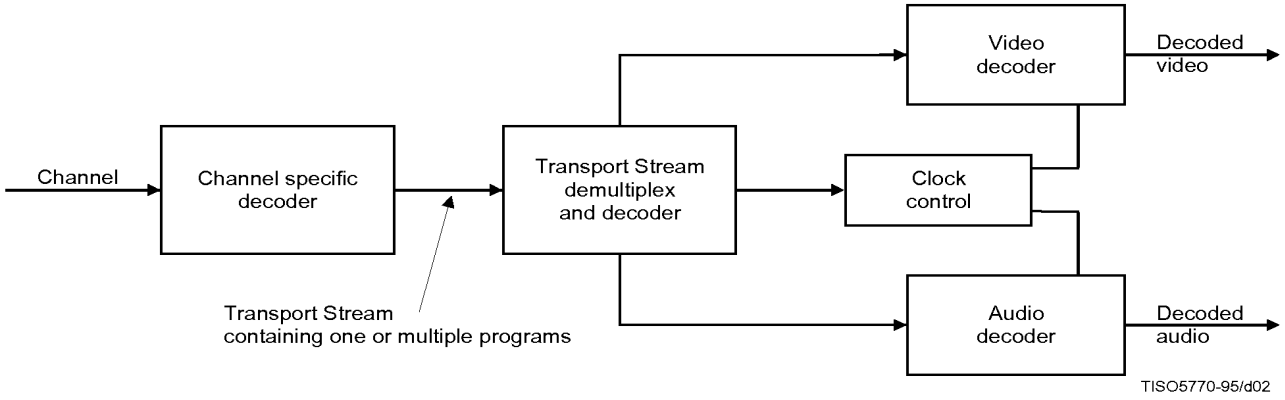


Figure Intro. 2 – Prototypical transport demultiplexing and decoding example

Figure Intro. 3 illustrates the second case, where a Transport Stream containing multiple programs is converted into a Transport Stream containing a single program. In this case the re-multiplexing operation may necessitate the correction of Program Clock Reference (PCR) values to account for changes in the PCR locations in the bit stream.

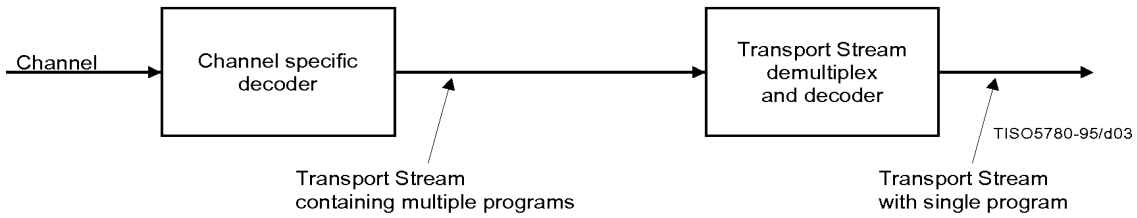


Figure Intro. 3 – Prototypical transport multiplexing example

 © ISO/IEC 2000 – All rights reserved

Figure Intro. 4 illustrates a case in which a multi-program Transport Stream is first demultiplexed and then converted into a Program Stream.

Figures Intro. 3 and Intro. 4 indicate that it is possible and reasonable to convert between different types and configurations of Transport Streams. There are specific fields defined in the **Transport Stream** and **Program Stream** syntax which facilitate the conversions illustrated. There is no requirement that specific implementations of demultiplexors or decoders include all of these functions.

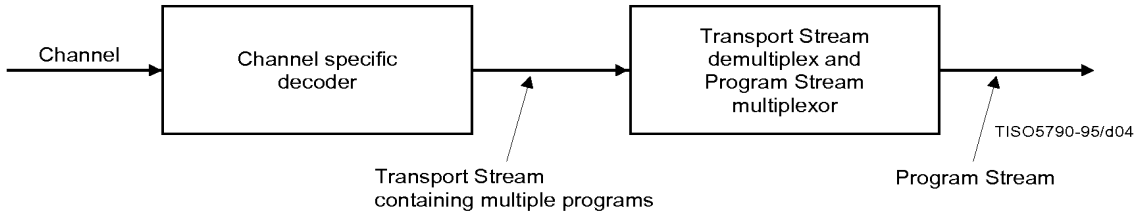


Figure Intro. 4 – Prototypical Transport Stream to Program Stream conversion

Intro. 2 Program Stream

The Program Stream is a stream definition which is tailored for communicating or storing one program of coded data and other data in environments where errors are very unlikely, and where processing of system coding, e.g. by software, is a major consideration.

Program Streams may be either fixed or variable rate. In either case, the constituent elementary streams may be either fixed or variable rate. The syntax and semantics constraints on the stream are identical in each case. The Program Stream rate is defined by the values and locations of the System Clock Reference (SCR) and mux_rate fields.

A prototypical audio/video Program Stream decoder system is depicted in Figure Intro. 5. The architecture is not unique – system decoder functions including decoder timing control might as equally well be distributed among elementary stream decoders and the channel specific decoder – but this figure is useful for discussion. The prototypical decoder design does not imply any normative requirement for the design of an Program Stream decoder. Indeed non-audio/video data is also allowed, but not shown.

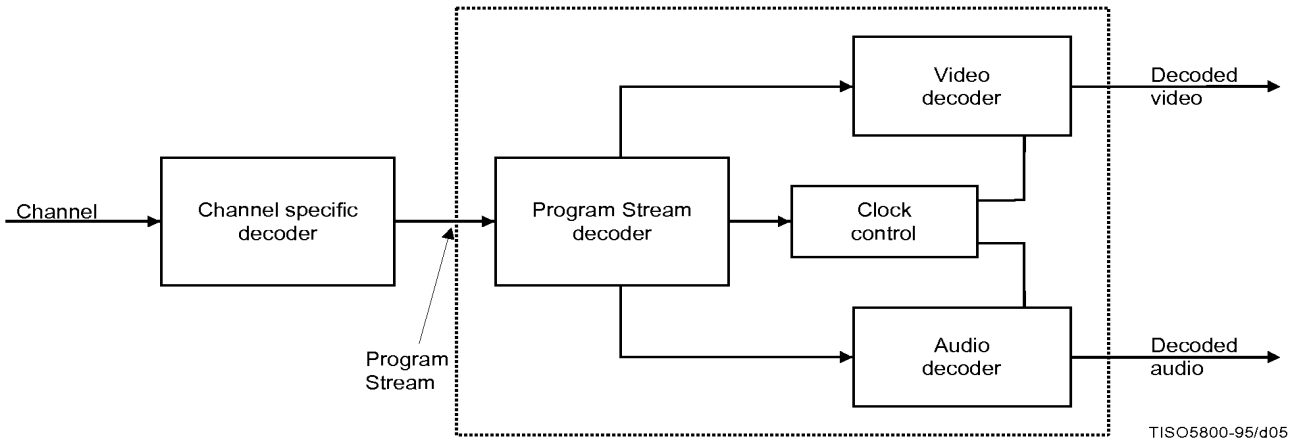


Figure Intro. 5 – Prototypical decoder for Program Streams

© ISO/IEC 2000 – All rights reserved

The prototypical decoder for Program Streams shown in Figure Intro. 5 is composed of System, Video, and Audio decoders conforming to Parts 1, 2, and 3, respectively, of ISO/IEC 13818. In this decoder, the multiplexed coded representation of one or more audio and/or video streams is assumed to be stored or communicated on some channel in some channel-specific format. The channel-specific format is not governed by this Recommendation | International Standard, nor is the channel-specific decoding part of the prototypical decoder.

The prototypical decoder accepts as input a Program Stream and relies on a Program Stream Decoder to extract timing information from the stream. The Program Stream Decoder demultiplexes the stream, and the elementary streams so produced serve as inputs to Video and Audio decoders, whose outputs are decoded video and audio signals. Included in the design, but not shown in the figure, is the flow of timing information among the Program Stream decoder, the Video and Audio decoders, and the channel-specific decoder. The Video and Audio decoders are synchronized with each other and with the channel using this timing information.

Program Streams are constructed in two layers: a system layer and a compression layer. The input stream to the Program Stream Decoder has a system layer wrapped about a compression layer. Input streams to the Video and Audio decoders have only the compression layer.

Operations performed by the prototypical decoder either apply to the entire Program Stream ("multiplex-wide operations"), or to individual elementary streams ("stream-specific operations"). The Program Stream system layer is divided into two sub-layers, one for multiplex-wide operations (the pack layer), and one for stream-specific operations (the PES packet layer).

Intro. 3 Conversion between Transport Stream and Program Stream

It may be possible and reasonable to convert between **Transport Streams** and **Program Streams** by means of PES packets. This results from the specification of **Transport Stream** and **Program Stream** as embodied in 2.4.1 and 2.5.1 of the normative requirements of this Recommendation | International Standard. PES packets may, with some constraints, be mapped directly from the payload of one multiplexed bit stream into the payload of another multiplexed bit stream. It is possible to identify the correct order of PES packets in a program to assist with this if the program_packet_sequence_counter is present in all PES packets.

Certain other information necessary for conversion, e.g. the relationship between elementary streams, is available in tables and headers in both streams. Such data, if available, shall be correct in any stream before and after conversion.

Intro. 4 Packetized Elementary Stream

Transport Streams and **Program Streams** are each logically constructed from PES packets, as indicated in the syntax definitions in 2.4.3.6. PES packets shall be used to convert between Transport Streams and Program Streams; in some cases the PES packets need not be modified when performing such conversions. PES packets may be much larger than the size of a Transport Stream packet.

A continuous sequence of PES packets of one elementary stream with one stream ID may be used to construct a PES Stream. When PES packets are used to form a PES stream, they shall include Elementary Stream Clock Reference (ESCR) fields and Elementary Stream Rate (ES_Rate) fields, with constraints as defined in 2.4.3.8. The PES stream data shall be contiguous bytes from the elementary stream in their original order. PES streams do not contain some necessary system information which is contained in Program Streams and Transport Streams. Examples include the information in the Pack Header, System Header, Program Stream Map, Program Stream Directory, Program Map Table, and elements of the Transport Stream packet syntax.

The PES Stream is a logical construct that may be useful within implementations of this Recommendation | International Standard; however, it is not defined as a stream for interchange and interoperability. Applications requiring streams containing only one elementary stream can use Program Streams or Transport Streams which each contain only one elementary stream. These streams contain all of the necessary system information. Multiple Program Streams or Transport Streams, each containing a single elementary stream, can be constructed with a common time base and therefore carry a complete program, i.e. with audio and video.

Intro. 5 Timing model

Systems, Video and Audio all have a timing model in which the end-to-end delay from the signal input to an encoder to the signal output from a decoder is a constant. This delay is the sum of encoding, encoder buffering, multiplexing, communication or storage, demultiplexing, decoder buffering, decoding, and presentation delays. As part of this timing model all video pictures and audio samples are presented exactly once, unless specifically coded to the contrary, and the inter-picture interval and audio sample rate are the same at the decoder as at the encoder. The system stream coding

 © ISO/IEC 2000 – All rights reserved

contains timing information which can be used to implement systems which embody constant end-to-end delay. It is possible to implement decoders which do not follow this model exactly; however, in such cases it is the decoder's responsibility to perform in an acceptable manner. The timing is embodied in the normative specifications of this Recommendation | International Standard, which must be adhered to by all valid bit streams, regardless of the means of creating them.

All timing is defined in terms of a common system clock, referred to as a System Time Clock. In the Program Stream this clock may have an exactly specified ratio to the video or audio sample clocks, or it may have an operating frequency which differs slightly from the exact ratio while still providing precise end-to-end timing and clock recovery.

In the Transport Stream the system clock frequency is constrained to have the exactly specified ratio to the audio and video sample clocks at all times; the effect of this constraint is to simplify sample rate recovery in decoders.

Intro. 6 Conditional access

Encryption and scrambling for conditional access to programs encoded in the Program and Transport Streams is supported by the system data stream definitions. Conditional access mechanisms are not specified here. The stream definitions are designed so that implementation of practical conditional access systems is reasonable, and there are some syntactical elements specified which provide specific support for such systems.

Intro. 7 Multiplex-wide operations

Multiplex-wide operations include the co-ordination of data retrieval of the channel, the adjustment of clocks, and the management of buffers. The tasks are intimately related. If the rate of data delivery of the channel is controllable, then data delivery may be adjusted so that decoder buffers neither overflow nor underflow; but if the data rate is not controllable, then elementary stream decoders must slave their timing to the data received from the channel to avoid overflow or underflow.

Program Streams are composed of packs whose headers facilitate the above tasks. Pack headers specify intended times at which each byte is to enter the Program Stream Decoder from the channel, and this target arrival schedule serves as a reference for clock correction and buffer management. The schedule need not be followed exactly by decoders, but they must compensate for deviations about it.

Similarly, Transport Streams are composed of Transport Stream packets with headers containing information which specifies the times at which each byte is intended to enter a Transport Stream Decoder from the channel. This schedule provides exactly the same function as that which is specified in the Program Stream.

An additional multiplex-wide operation is a decoder's ability to establish what resources are required to decode a Transport Stream or Program Stream. The first pack of each Program Stream conveys parameters to assist decoders in this task. Included, for example, are the stream's maximum data rate and the highest number of simultaneous video channels. The Transport Stream likewise contains globally useful information.

The Transport Stream and Program Stream each contain information which identifies the pertinent characteristics of, and relationships between, the elementary streams which constitute each program. Such information may include the language spoken in audio channels, as well as the relationship between video streams when multi-layer video coding is implemented.

Intro. 8 Individual stream operations (PES Packet Layer)

The principal stream-specific operations are:

1) demultiplexing; and

2) synchronizing playback of multiple elementary streams.

Intro. 8.1 Demultiplexing

On encoding, Program Streams are formed by multiplexing elementary streams, and Transport Streams are formed by multiplexing elementary streams, Program Streams, or the contents of other Transport Streams. Elementary streams may include private, reserved, and padding streams in addition to audio and video streams. The streams are temporally subdivided into packets, and the packets are serialized. A PES packet contains coded bytes from one and only one elementary stream.

© ISO/IEC 2000 – All rights reserved

In the Program Stream both fixed and variable packet lengths are allowed subject to constraints as specified in 2.5.1 and 2.5.2. For Transport Streams the packet length is 188 bytes. Both fixed and variable PES packet lengths are allowed, and will be relatively long in most applications.

On decoding, demultiplexing is required to reconstitute elementary streams from the multiplexed Program Stream or Transport Stream. Stream_id codes in Program Stream packet headers, and Packet ID codes in the Transport Stream make this possible.

Intro. 8.2 Synchronization

Synchronization among multiple elementary streams is accomplished with Presentation Time Stamps (PTS) in the Program Stream and Transport streams. Time stamps are generally in units of 90 kHz, but the System Clock Reference (SCR), the Program Clock Reference (PCR) and the optional Elementary Stream Clock Reference (ESCR) have extensions with a resolution of 27 MHz. Decoding of N-elementary streams is synchronized by adjusting the decoding of streams to a common master time base rather than by adjusting the decoding of one stream to match that of another. The master time base may be one of the N-decoders' clocks, the data source's clock, or it may be some external clock.

Each program in a Transport Stream, which may contain multiple programs, may have its own time base. The time bases of different programs within a Transport Stream may be different.

Because PTSs apply to the decoding of individual elementary streams, they reside in the PES packet layer of both the Transport Streams and Program Streams. End-to-end synchronization occurs when encoders save time stamps at capture time, when the time stamps propagate with associated coded data to decoders, and when decoders use those time stamps to schedule presentations.

Synchronization of a decoding system with a channel is achieved through the use of the SCR in the Program Stream and by its analogue, the PCR, in the Transport Stream. The SCR and PCR are time stamps encoding the timing of the bit stream itself, and are derived from the same time base used for the audio and video PTS values from the same program. Since each program may have its own time base, there are separate PCR fields for each program in a Transport Stream containing multiple programs. In some cases it may be possible for programs to share PCR fields. Refer to 2.4.4, Program Specific Information (PSI), for the method of identifying which PCR is associated with a program. A program shall have one and only one PCR time base associated with it.

Intro. 8.3 Relation to compression layer

The PES packet layer is independent of the compression layer in some senses, but not in all. It is independent in the sense that PES packet payloads need not start at compression layer start codes, as defined in Parts 2 and 3 of ISO/IEC 13818. For example, video start codes may occur anywhere within the payload of a PES packet, and start codes may be split by a PES packet header. However, time stamps encoded in PES packet headers apply to presentation times of compression layer constructs (namely, presentation units). In addition, when the elementary stream data conforms to ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 13818-3, the PES_packet_data_bytes shall be byte aligned to the bytes of this Recommendation | International Standard.

Intro. 9 System reference decoder

Part 1 of ISO/IEC 13818 employs a "System Target Decoder" (STD), one for Transport Streams (refer to 2.4.2) referred to as "Transport System Target Decoder" (T-STD) and one for Program Streams (refer to 2.5.2) referred to as "Program System Target Decoder" (P-STD), to provide a formalism for timing and buffering relationships. Because the STD is parameterized in terms of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 fields (for example, buffer sizes) each elementary stream leads to its own parameterization of the STD. Encoders shall produce bit streams that meet the appropriate STD's constraints. Physical decoders may assume that a stream plays properly on its STD. The physical decoder must compensate for ways in which its design differs from that of the STD.

Intro. 10 Applications

The streams defined in this Recommendation | International Standard are intended to be as useful as possible to a wide variety of applications. Application developers should select the most appropriate stream.

Modern data communications networks may be capable of supporting ITU-T Rec. H.222.0 | ISO/IEC 13818-1 video and ISO/IEC 13818 audio. A real time transport protocol is required. The Program Stream may be suitable for transmission on such networks.

 © ISO/IEC 2000 – All rights reserved

The Program Stream is also suitable for multimedia applications on CD-ROM. Software processing of the Program Stream may be appropriate.

The Transport Stream may be more suitable for error-prone environments, such as those used for distributing compressed bit-streams over long distance networks and in broadcast systems.

Many applications require storage and retrieval of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstreams on various Digital Storage Media (DSM). A Digital Storage Media Command and Control (DSM-CC) protocol is specified in Annex B and Part 6 of ISO/IEC 13818 in order to facilitate the control of such media.

© ISO/IEC 2000 – All rights reserved

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – GENERIC CODING OF MOVING PICTURES AND ASSOCIATED AUDIO INFORMATION: SYSTEMS

SECTION 1 – GENERAL

1.1 Scope

This Recommendation | International Standard specifies the system layer of the coding. It was developed principally to support the combination of the video and audio coding methods defined in Parts 2 and 3 of ISO/IEC 13818. The system layer supports five basic functions:

1) the synchronization of multiple compressed streams on decoding;

2) the interleaving of multiple compressed streams into a single stream;

3) the initialization of buffering for decoding start up;

4) continuous buffer management; and

5) time identification.

An ITU-T Rec. H.222.0 | ISO/IEC 13818-1 multiplexed bit stream is either a **Transport Stream** or a **Program Stream**. Both streams are constructed from **PES packets** and packets containing other necessary information. Both stream types support multiplexing of video and audio compressed streams from one program with a common time base. The **Transport Stream** additionally supports the multiplexing of video and audio compressed streams from multiple programs with independent time bases. For almost error-free environments the **Program Stream** is generally more appropriate, supporting software processing of program information. The **Transport Stream** is more suitable for use in environments where errors are likely.

An ITU-T Rec. H.222.0 | ISO/IEC 13818-1 multiplexed bit stream, whether a Transport Stream or a Program Stream, is constructed in two layers: the outermost layer is the system layer, and the innermost is the compression layer. The system layer provides the functions necessary for using one or more compressed data streams in a system. The video and audio parts of this Specification define the compression coding layer for audio and video data. Coding of other types of data is not defined by this Specification, but is supported by the system layer provided that the other types of data adhere to the constraints defined in 2.7.

1.2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

1.2.1 Identical Recommendations | International Standards

– ITU-T Recommendation H.262 (2000) | ISO/IEC 13818-2:2000, *Information technology – Generic coding of moving pictures and associated audio information: Video.*

1.2.2 Paired Recommendations | International Standards equivalent in technical content

– ITU-T Recommendation T.171 (1996), *Protocols for interactive audiovisual services: coded representation of multimedia and hypermedia objects.*

ISO/IEC 13522-1:1997, *Information technology – Coding of multimedia and hypermedia information – Part 1: MHEG object representation – Base notation (ASN.1).*

1.2.3 Additional references

– ISO 639-2:1998, *Codes for the representation of names of languages – Part 2: Alpha-3 code.*

– ISO 8859-1:1987, *Information processing – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1.*

– ISO/IEC 11172-1:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 1: Systems.*

– ISO/IEC 11172-2:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video.*

– ISO/IEC 11172-3:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio.*

– ISO/IEC 13818-3:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio.*

– ISO/IEC 13818-6:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 6: Extensions for DSM-CC.*

– ISO/IEC 13818-7:1997, *Information technology – Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC)*

– ISO/IEC 14496-1:1999, *Information technology – Coding of audio-visual objects – Part 1: Systems*

– ISO/IEC 14496-2:1999, *Information technology – Coding of audio-visual objects – Part 2: Visual*

– ISO/IEC 14496-3:1999, *Information technology – Coding of audio-visual objects – Part 3: Audio*

– ISO/IEC 14496-3:1999/Amd.1:2000, *Information technology – Coding of audio-visual objects – Part 3: Audio, Amendment 1: Audio extensions*

– Recommendation ITU-R BT.601.3, *Encoding parameters of digital television for studios.*

– Recommendation ITU-R BT.470-2, *Television systems.*

– Recommendation ITU-R BR.648, *Digital recording of audio signals.*

– Report ITU-R BO.955.2, *Satellite sound broadcasting of vehicular, portable, and fixed receivers in the range 500-3000 MHz.*

– CCITT Recommendation J.17 (1988), *Pre-emphasis used on sound-programme circuits.*

– IEEE Standard 1180:1990, *Standard Specification for the Implementations of 8 by 8 Inverse Discrete Cosine Transform.*

– IEC Publication 60908:1987, *Compact disc digital audio system.*

SECTION 2 – TECHNICAL ELEMENTS

2.1 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply. If specific to a Part, this is parenthetically noted.

2.1.1 access unit (system): A coded representation of a presentation unit. In the case of audio, an access unit is the coded representation of an audio frame.

In the case of video, an access unit includes all the coded data for a picture, and any stuffing that follows it, up to but not including the start of the next access unit. If a picture is not preceded by a group_start_code or a sequence_header_code, the access unit begins with the picture start code. If a picture is preceded by a group_start_code and/or a sequence_header_code, the access unit begins with the first byte of the first of these start codes. If it is the last picture preceding a sequence_end_code in the bitstream, all bytes between the last byte of the coded picture and the sequence_end_code (including the sequence_end_code) belong to the access unit.

2.1.2 bitrate: The rate at which the compressed bit stream is delivered from the channel to the input of a decoder.

2.1.3 byte aligned: A bit in a coded bit stream is byte-aligned if its position is a multiple of 8-bits from the first bit in the stream.

2.1.4 channel: A digital medium that stores or transports an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream.

2.1.5 coded representation: A data element as represented in its encoded form.

2.1.6 compression: Reduction in the number of bits used to represent an item of data.

2.1.7 constant bitrate: Operation where the bitrate is constant from start to finish of the compressed bit stream.

2.1.8 constrained system parameter stream; CSPS (system): A Program Stream for which the constraints defined in 2.7.9 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 apply.

2.1.9 CRC: The Cyclic Redundancy Check to verify the correctness of data.

2.1.10 data element: An item of data as represented before encoding and after decoding.

2.1.11 decoded stream: The decoded reconstruction of a compressed bit stream.

2.1.12 decoder: An embodiment of a decoding process.

2.1.13 decoding (process): The process defined in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 that reads an input coded bit stream and outputs decoded pictures or audio samples.

2.1.14 decoding time-stamp; DTS (system): A field that may be present in a PES packet header that indicates the time that an access unit is decoded in the system target decoder.

2.1.15 digital storage media (DSM): A digital storage or transmission device or system.

2.1.16 DSM-CC: Digital storage media command and control.

2.1.17 entitlement control message (ECM): Entitlement Control Messages are private conditional access information which specify control words and possibly other, typically stream-specific, scrambling and/or control parameters.

2.1.18 entitlement management message (EMM): Entitlement Management Messages are private conditional access information which specify the authorization levels or the services of specific decoders. They may be addressed to single decoders or groups of decoders.

2.1.19 editing: The process by which one or more compressed bit streams are manipulated to produce a new compressed bit stream. Edited bit streams meet the same requirements as streams which are not edited.

2.1.20 elementary stream; ES (system): A generic term for one of the coded video, coded audio or other coded bit streams in PES packets. One elementary stream is carried in a sequence of PES packets with one and only one stream_id.

2.1.21 Elementary Stream Clock Reference; ESCR (system): A time stamp in the PES Stream from which decoders of PES streams may derive timing.

2.1.22 encoder: An embodiment of an encoding process.

2.1.23 encoding (process): A process, not specified in ITU-T Rec. H.222.0 | ISO/IEC 13818-1, that reads a stream of input pictures or audio samples and produces a coded bit stream conforming to ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

2.1.24 entropy coding: Variable length lossless coding of the digital representation of a signal to reduce redundancy.

2.1.25 event: An event is defined as a collection of elementary streams with a common time base, an associated start time, and an associated end time.

2.1.26 fast forward playback (video): The process of displaying a sequence, or parts of a sequence, of pictures in display-order faster than real-time.

2.1.27 forbidden: The term "forbidden", when used in the clauses of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 defining the coded bit stream, indicates that the value specified shall never be used.

2.1.28 (multiplexed) stream (system): A bit stream composed of 0 or more elementary streams combined in a manner that conforms to ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

2.1.29 layer (video and systems): One of the levels in the data hierarchy of the video and system specifications defined in Parts 1 and 2 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

2.1.30 pack (system): A pack consists of a pack header followed by zero or more packets. It is a layer in the system coding syntax described in 2.5.3.3 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

2.1.31 packet data (system): Contiguous bytes of data from an elementary stream present in a packet.

2.1.32 packet identifier; PID (system): A unique integer value used to identify elementary streams of a program in a single or multi-program Transport Stream as described in 2.4.3 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

2.1.33 padding (audio): A method to adjust the average length of an audio frame in time to the duration of the corresponding PCM samples, by conditionally adding a slot to the audio frame.

2.1.34 payload: Payload refers to the bytes which follow the header bytes in a packet. For example, the payload of some Transport Stream packets includes a PES_packet_header and its PES_packet_data_bytes, or pointer_field and PSI sections, or private data; but a PES_packet_payload consists of only PES_packet_data_bytes. The Transport Stream packet header and adaptation fields are not payload.

2.1.35 PES (system): An abbreviation for Packetized Elementary Stream.

2.1.36 PES packet (system): The data structure used to carry elementary stream data. A PES packet consists of a PES packet header followed by a number of contiguous bytes from an elementary data stream. It is a layer in the system coding syntax described in 2.4.3.6 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

2.1.37 PES packet header (system): The leading fields in a PES packet up to and not including the PES_packet_data_byte fields, where the stream is not a padding stream. In the case of a padding stream the PES packet header is similarly defined as the leading fields in a PES packet up to and not including padding_byte fields.

2.1.38 PES Stream (system): A PES Stream consists of PES packets, all of whose payloads consist of data from a single elementary stream, and all of which have the same stream_id. Specific semantic constraints apply. Refer to Intro. 4 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

2.1.39 presentation time-stamp; PTS (system): A field that may be present in a PES packet header that indicates the time that a presentation unit is presented in the system target decoder.

2.1.40 presentation unit; PU (system): A decoded Audio Access Unit or a decoded picture.

2.1.41 program (system): A program is a collection of program elements. Program elements may be elementary streams. Program elements need not have any defined time base; those that do, have a common time base and are intended for synchronized presentation.

2.1.42 Program Clock Reference; PCR (system): A time stamp in the Transport Stream from which decoder timing is derived.

2.1.43 program element (system): A generic term for one of the elementary streams or other data streams that may be included in a program.

2.1.44 Program Specific Information; PSI (system): PSI consists of normative data which is necessary for the demultiplexing of Transport Streams and the successful regeneration of programs and is described in 2.4.4 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1. An example of privately defined PSI data is the non-mandatory network information table.

2.1.45 random access: The process of beginning to read and decode the coded bit stream at an arbitrary point.

2.1.46 reserved: The term "reserved", when used in the clauses defining the coded bit stream, indicates that the value may be used in the future for ISO defined extensions. Unless otherwise specified within ITU-T Rec. H.222.0 | ISO/IEC 13818-1, all reserved bits shall be set to '1'.

2.1.47 scrambling (system): The alteration of the characteristics of a video, audio or coded data stream in order to prevent unauthorized reception of the information in a clear form. This alteration is a specified process under the control of a conditional access system.

2.1.48 source stream: A single non-multiplexed stream of samples before compression coding.

2.1.49 splicing (system): The concatenation, performed on the system level, of two different elementary streams. The resulting system stream conforms totally to ITU-T Rec. H.222.0 | ISO/IEC 13818-1. The splice may result in discontinuities in timebase, continuity counter, PSI, and decoding.

2.1.50 start codes (system): 32-bit codes embedded in the coded bit stream. They are used for several purposes including identifying some of the layers in the coding syntax. Start codes consist of a 24-bit prefix (0x000001) and an 8-bit stream_id as shown in Table 2-18 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

2.1.51 STD input buffer (system): A first-in first-out buffer at the input of a system target decoder for storage of compressed data from elementary streams before decoding.

2.1.52 still picture: A coded still picture consists of a video sequence containing exactly one coded picture which is intra-coded. This picture has an associated PTS and the presentation time of succeeding pictures, if any, is later than that of the still picture by at least two picture periods.

2.1.53 system header (system): The system header is a data structure defined in 2.5.3.5 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 that carries information summarizing the system characteristics of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Stream.

2.1.54 System Clock Reference; SCR (system): A time stamp in the Program Stream from which decoder timing is derived.

2.1.55 system target decoder; STD (system): A hypothetical reference model of a decoding process used to define the semantics of an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 multiplexed bit stream.

2.1.56 time-stamp (system): A term that indicates the time of a specific action such as the arrival of a byte or the presentation of a Presentation Unit.

2.1.57 Transport Stream packet header (system): The leading fields in a Transport Stream packet, up to and including the continuity_counter field.

2.1.58 variable bitrate: An attribute of Transport Streams or Program Streams wherein the rate of arrival of bytes at the input to a decoder varies with time.

2.2 Symbols and abbreviations

The mathematical operators used to describe this Recommendation | International Standard are similar to those used in the C-programming language. However, integer division with truncation and rounding are specifically defined. The bitwise operators are defined assuming two's-complement representation of integers. Numbering and counting loops generally begin from 0.

2.2.1 Arithmetic operators

+	Addition
−	Subtraction (as a binary operator) or negation (as a unary operator)
++	Increment
−−	Decrement
* or ×	Multiplication
^	Power
/	Integer division with truncation of the result toward 0. For example, 7/4 and −7/−4 are truncated to 1 and −7/4 and 7/−4 are truncated to −1.
//	Integer division with rounding to the nearest integer. Half-integer values are rounded away from 0 unless otherwise specified. For example 3//2 is rounded to 2, and −3//2 is rounded to −2.
DIV	Integer division with truncation of the result towards −∞

%	Modulus operator. Defined only for positive numbers.
Sign( )	$Sign(x) = \begin{cases} 1 & x > 0 \\ 0 & x == 0 \\ -1 & x < 0 \end{cases}$
NINT( )	Nearest integer operator. Returns the nearest integer value to the real-valued argument. Half-integer values are rounded away from 0.
sin	Sine
cos	Cosine
exp	Exponential
√	Square root
$\log_{10}$	Logarithm to base ten
$\log_e$	Logarithm to base e

2.2.2 Logical operators

||	Logical OR
&&	Logical AND
!	Logical NOT

2.2.3 Relational operators

>	Greater than
≥	Greater than or equal to
<	Less than
≤	Less than or equal to
==	Equal to
!=	Not equal to
max [,...,]	The maximum value in the argument list
min [,...,]	The minimum value in the argument list

2.2.4 Bitwise operators

&	AND
|	OR
>>	Shift right with sign extension
<<	Shift left with 0 fill

2.2.5 Assignment

=	Assignment operator

2.2.6 Mnemonics

The following mnemonics are defined to describe the different data types used in the coded bit-stream.

bslbf	Bit string, left bit first, where "left" is the order in which bit strings are written in this Recommendation | International Standard. Bit strings are written as a string of 1s and 0s within single quote marks, e.g. '1000 0001'. Blanks within a bit string are for ease of reading and have no significance.
ch	Channel
gr	Granule of 3 * 32 sub-band samples in audio Layer II, 18 * 32 sub-band samples in audio Layer III.
main_data	The main_data portion of the bit stream contains the scale factors, Huffman encoded data, and ancillary information.

main_data_beg	This gives the location in the bit stream of the beginning of the main_data for the frame. The location is equal to the ending location of the previous frame's main_data plus 1 bit. It is calculated from the main_data_end value of the previous frame.
part2_length	This value contains the number of main_data bits used for scale factors
rpchof	Remainder polynomial coefficients, highest order first
sb	Sub-band
scfsi	Scalefactor selector information
switch_point_l	Number of scalefactor band (long block scalefactor band) from which point on window switching is used
switch_point_s	Number of scalefactor band (short block scalefactor band) from which point on window switching is used
tcimsbf	Two's complement integer, msb (sign) bit first
uimsbf	Unsigned integer, most significant bit first
vlclbf	Variable length code, left bit first, where "left" refers to the order in which the variable length codes are written
window	Number of actual time slot in case of block_type == 2, $0 \leq$ window ≤ 2.

The byte order of multi-byte words is most significant byte first.

2.2.7 Constants

π	3.14159265359
e	2.71828182845

2.3 Method of describing bit stream syntax

The bit streams retrieved by the decoder are described in 2.4.1 and 2.5.1. Each data item in the bit stream is in bold type. It is described by its name, its length in bits, and a mnemonic for its type and order of transmission.

The action caused by a decoded data element in a bit stream depends on the value of that data element and on data elements previously decoded. The decoding of the data elements and definition of the state variables used in their decoding are described in the clauses containing the semantic description of the syntax. The following constructs are used to express the conditions when data elements are present, and are in normal type.

Note this syntax uses the "C"-code convention that a variable or expression evaluating to a non-zero value is equivalent to a condition that is true:

while (condition) {
  data_element
  . . .
}
	If the condition is true, then the group of data elements occurs next in the data stream. This repeats until the condition is not true.

do {
  data_element
  . . .
}
while (condition)
	The data element always occurs at least once. The data element is repeated until the condition is not true.

if (condition) {
  data_element
  . . .
}
	If the condition is true, then the first group of data elements occurs next in the data stream.

else {
  data_element
  . . .
}
	If the condition is not true, then the second group of data elements occurs next in the data stream.

for (i = 0; i < n; i++) {
  data_element
  . . .
}
	The group of data elements occurs n times. Conditional constructs within the group of data elements may depend on the value of the loop control variable i, which is set to zero for the first occurrence, incremented to 1 for the second occurrence, and so forth.

As noted, the group of data elements may contain nested conditional constructs. For compactness, the {} are omitted when only one data element follows:

data_element [] data_element [] is an array of data. The number of data elements is indicated by the context.

data_element [n] data_element [n] is the n+1th element of an array of data.

data_element [m][n] data_element [m][n] is the m+1,n+1th element of a two-dimensional array of data.

data_element [l][m][n] data_element [l][m][n] is the l+1,m+1,n+1th element of a three-dimensional array of data.

data_element [m..n] is the inclusive range of bits between bit m and bit n in the data_element.

While the syntax is expressed in procedural terms, it should not be assumed that either Figure 2-1 or Figure 2-2 implements a satisfactory decoding procedure. In particular, they define a correct and error-free input bitstream. Actual decoders must include a means to look for start codes and sync bytes (Transport Stream) in order to begin decoding correctly, and to identify errors, erasures or insertions while decoding. The methods to identify these situations, and the actions to be taken, are not standardized.

2.4 Transport Stream bitstream requirements

2.4.1 Transport Stream coding structure and parameters

The ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Transport Stream coding layer allows one or more programs to be combined into a single stream. Data from each elementary stream are multiplexed together with information that allows synchronized presentation of the elementary streams within a program.

A Transport Stream consists of one or more programs. Audio and video elementary streams consist of access units.

Elementary Stream data is carried in PES packets. A PES packet consists of a PES packet header followed by packet data. PES packets are inserted into Transport Stream packets. The first byte of each PES packet header is located at the first available payload location of a Transport Stream packet.

The PES packet header begins with a 32-bit start-code that also identifies the stream or stream type to which the packet data belongs. The PES packet header may contain decoding and presentation time stamps (DTS and PTS). The PES packet header also contains other optional fields. The PES packet data field contains a variable number of contiguous bytes from one elementary stream.

Transport Stream packets begin with a 4-byte prefix, which contains a 13-bit Packet ID (PID), defined in Table 2-2. The PID identifies, via the Program Specific Information (PSI) tables, the contents of the data contained in the Transport Stream packet. Transport Stream packets of one PID value carry data of one and only one elementary stream.

The PSI tables are carried in the Transport Stream. There are four PSI tables:

- Program Association Table;
- Program Map Table;
- Conditional Access Table;
- Network Information Table.

These tables contain the necessary and sufficient information to demultiplex and present programs. The Program Map Table, in Table 2-28, specifies, among other information, which PIDs, and therefore which elementary streams are associated to form each program. This table also indicates the PID of the Transport Stream packets which carry the PCR for each program. The Conditional Access Table shall be present if scrambling is employed. The Network Information Table is optional and its contents are not specified by this Recommendation | International Standard.

Transport Stream packets may be null packets. Null packets are intended for padding of Transport Streams. They may be inserted or deleted by re-multiplexing processes and, therefore, the delivery of the payload of null packets to the decoder cannot be assumed.

This Recommendation | International Standard does not specify the coded data which may be used as part of conditional access systems. This Specification does, however, provide mechanisms for program service providers to transport and identify this data for decoder processing, and to reference correctly data which are specified by this Specification. This type of support is provided both through Transport Stream packet structures and in the conditional access table (refer to Table 2-27 of the PSI).

2.4.2 Transport Stream system target decoder

The semantics of the Transport Stream specified in 2.4.3 and the constraints on these semantics specified in 2.7 require exact definitions of byte arrival and decoding events and the times at which these occur. The definitions needed are set out in this Recommendation | International Standard using a hypothetical decoder known as the Transport Stream System Target Decoder (T-STD). Informative Annex D contains further explanation of the T-STD.

The T-STD is a conceptual model used to define these terms precisely and to model the decoding process during the construction or verification of Transport Streams. The T-STD is defined only for this purpose. There are three types of decoders in the T-STD: video, audio, and systems. Figure 2-1 illustrates an example. Neither the architecture of the T-STD nor the timing described precludes uninterrupted, synchronized play-back of Transport Streams from a variety of decoders with different architectures or timing schedules.

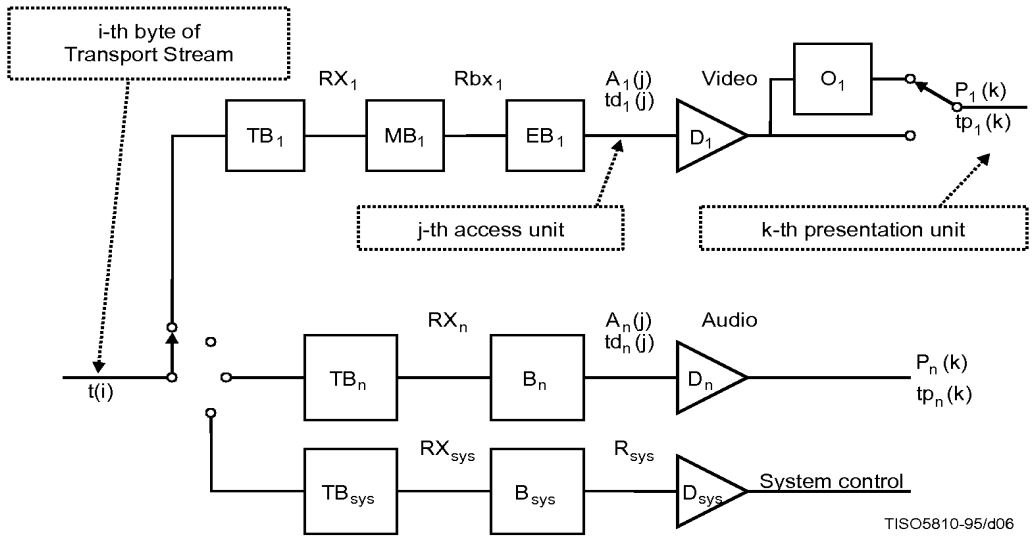


Figure 2-1 – Transport Stream system target decoder notation

The following notation is used to describe the Transport Stream system target decoder and is partially illustrated in Figure 2-1 above.

i, i′, i″ are indices to bytes in the Transport Stream. The first byte has index 0.

j is an index to access units in the elementary streams.

k, k′, k″ are indices to presentation units in the elementary streams.

n is an index to the elementary streams.

p is an index to Transport Stream packets in the Transport Stream.

t(i) indicates the time in seconds at which the i-th byte of the Transport Stream enters the system target decoder. The value t(0) is an arbitrary constant.

PCR(i) is the time encoded in the PCR field measured in units of the period of the 27 MHz system clock where i is the byte index of the final byte of the program_clock_reference_base field.

$A_n(j)$ is the j-th access unit in elementary stream n. $A_n(j)$ is indexed in decoding order.

$td_n(j)$ is the decoding time, measured in seconds, in the system target decoder of the j-th access unit in elementary stream n.

$P_n(k)$ is the k-th presentation unit in elementary stream n. $P_n(k)$ results from decoding $A_n(j)$. $P_n(k)$ is indexed in presentation order.

$tp_n(k)$ is the presentation time, measured in seconds, in the system target decoder of the k-th presentation unit in elementary stream n.

t	is time measured in seconds.
$F_n(t)$	is the fullness, measured in bytes, of the system target decoder input buffer for elementary stream n at time t.
B_n	is the main buffer for elementary stream n. It is present only for audio elementary streams.
BS_n	is the size of buffer, B_n, measured in bytes.
B_{sys}	is the main buffer in the system target decoder for system information for the program that is in the process of being decoded.
BS_{sys}	is the size of B_{sys}, measured in bytes.
MB_n	is the multiplexing buffer, for elementary stream n. It is present only for video elementary streams.
MBS_n	is the size of MB_n, measured in bytes.
EB_n	is the elementary stream buffer for elementary stream n. It is present only for video elementary streams.
EBS_n	is the size of the elementary stream buffer EB_n, measured in bytes.
TB_{sys}	is the transport buffer for system information for the program that is in the process of being decoded.
TBS_{sys}	is the size of TB_{sys}, measured in bytes.
TB_n	is the transport buffer for elementary stream n.
TBS_n	is the size of TB_n, measured in bytes.
D_{sys}	is the decoder for system information in Program Stream n.
D_n	is the decoder for elementary stream n.
O_n	is the re-order buffer for video elementary stream n.
R_{sys}	is the rate at which data are removed from B_{sys}.
Rx_n	is the rate at which data are removed from TB_n.
Rbx_n	is the rate at which PES packet payload data are removed from MB_n when the leak method is used. Defined only for video elementary streams.
$Rbx_n(j)$	is the rate at which PES packet payload data are removed from MB_n when the vbv_delay method is used. Defined only for video elementary streams.
Rx_{sys}	The rate at which data are removed from TB_{sys}.
R_{es}	The video elementary stream rate coded in a sequence header.

2.4.2.1 System clock frequency

Timing information referenced in the T-STD is carried by several data fields defined in this Specification. Refer to 2.4.3.4 and 2.4.3.6. In PCR fields this information is coded as the sampled value of a program's system clock. The PCR fields are carried in the adaptation field of the Transport Stream packets with a PID value equal to the PCR_PID defined in the TS_program_map_section of the program being decoded.

Practical decoders may reconstruct this clock from these values and their respective arrival times. The following are minimum constraints which apply to the program's system clock frequency as represented by the values of the PCR fields when they are received by a decoder.

The value of the system clock frequency is measured in Hz and shall meet the following constraints:

$$27\,000\,000 - 810 \leq \text{system\_clock\_frequency} \leq 27\,000\,000 + 810$$

$$\text{rate of change of system\_clock\_frequency with time} \leq 75 \times 10^{-3}\ \text{Hz/s}$$

NOTE – Sources of coded data should follow a tighter tolerance in order to facilitate compliant operation of consumer recorders and playback equipment.

A program's system_clock_frequency may be more accurate than required. Such improved accuracy may be transmitted to the decoder via the System clock descriptor described in 2.6.20.

Bit rates defined in this Specification are measured in terms of system_clock_frequency. For example, a bit rate of 27 000 000 bits per second in the T-STD would indicate that one byte of data is transferred every eight (8) cycles of the system clock.

The notation "system_clock_frequency" is used in several places in this Specification to refer to the frequency of a clock meeting these requirements. For notational convenience, equations in which PCR, PTS, or DTS appear, lead to values of time which are accurate to some integral multiple of (300×2^{33}/system_clock_frequency) seconds. This is due to the encoding of PCR timing information as 33 bits of 1/300 of the system clock frequency plus 9 bits for the remainder, and encoding as 33 bits of the system clock frequency divided by 300 for PTS and DTS.

2.4.2.2 Input to the Transport Stream system target decoder

Input to the Transport Stream System Target Decoder (T-STD) is a Transport Stream. A Transport Stream may contain multiple programs with independent time bases. However, the T-STD decodes only one program at a time. In the T-STD model all timing indications refer to the time base of that program.

Data from the Transport Stream enters the T-STD at a piecewise constant rate. The time t(i) at which the i-th byte enters the T-STD is defined by decoding the program clock reference (PCR) fields in the input stream, encoded in the Transport Stream packet adaptation field of the program to be decoded and by counting the bytes in the complete Transport Stream between successive PCRs of that program. The PCR field (see equation 2-1) is encoded in two parts: one, in units of the period of 1/300 times the system clock frequency, called program_clock_reference_base (see equation 2-2), and one in units of the system clock frequency called program_clock_reference_extension (see equation 2-3). The values encoded in these are computed by PCR_base(i) (see equation 2-2) and PCR_ext(i) (see equation 2-3) respectively. The value encoded in the PCR field indicates the time t(i), where i is the index of the byte containing the last bit of the program_clock_reference_base field.

Specifically:

$$PCR(i) = PCR\_base(i) \times 300 + PCR\_ext(i) \tag{2-1}$$

where:

$$PCR\_base(i) = ((system\_clock\_frequency \times t(i)) \; DIV \; 300) \; \% \; 2^{33} \tag{2-2}$$

$$PCR\_ext(i) = ((system\_clock\_frequency \times t(i)) \; DIV \; 1) \; \% \; 300 \tag{2-3}$$

For all other bytes the input arrival time, t(i) shown in equation 2-4 below, is computed from PCR(i″) and the transport rate at which data arrive, where the transport rate is determined as the number of bytes in the Transport Stream between the bytes containing the last bit of two successive program_clock_reference_base fields of the same program divided by the difference between the time values encoded in these same two PCR fields.

$$t(i) = \frac{PCR(i'')}{system\_clock\_frequency} + \frac{i - i''}{transport\_rate\ (i)} \tag{2-4}$$

where:

i is the index of any byte in the Transport Stream for $i'' < i < i'$.

i″ is the index of the byte containing the last bit of the most recent program_clock_reference_base field applicable to the program being decoded.

PCR(i″) is the time encoded in the program clock reference base and extension fields in units of the system clock.

The transport rate is given by

$$transport\_rate(i) = \frac{((i' - i'') \times system\_clock\_frequency)}{PCR(i') - PCR(i'')} \tag{2-5}$$

where

i′ is the index of the byte containing the last bit of the immediately following program_clock_reference_base field applicable to the program being decoded.

NOTE – $i'' < i \leq i'$

In the case of a timebase discontinuity, indicated by the discontinuity_indicator in the transport packet adaptation field, the definition given in equation 2-4 and equation 2-5 for the time of arrival of bytes at the input to the T-STD is not applicable between the last PCR of the old timebase and the first PCR of the new timebase. In this case the time of arrival of these bytes is determined according to equation 2-4 with the modification that the transport rate used is that applicable between the last and next to last PCR of the old timebase.

A tolerance is specified for the PCR values. The PCR tolerance is defined as the maximum inaccuracy allowed in received PCRs. This inaccuracy may be due to imprecision in the PCR values or to PCR modification during re-multiplexing. It does not include errors in packet arrival time due to network jitter or other causes. The PCR tolerance is ± 500 ns.

In the T-STD model, the inaccuracy will be reflected as an inaccuracy in the calculated transport rate using equation 2-5.

Transport Streams with multiple programs and variable rate

Transport Streams may contain multiple programs which have independent time bases. Separate sets of PCRs, as indicated by the respective PCR_PID values, are required for each such independent program, and therefore the PCRs cannot be co-located. The Transport Stream rate is piecewise constant for the program entering the T-STD. Therefore, if the Transport Stream rate is variable it can only vary at the PCRs of the program under consideration. Since the PCRs, and therefore the points in the transport Stream where the rate varies, are not co-located, the rate at which the Transport Stream enters the T-STD would have to differ depending on which program is entering the T-STD. Therefore, it is not possible to construct a consistent T-STD delivery schedule for an entire Transport Stream when that Transport Stream contains multiple programs with independent time bases and the rate of the Transport Stream is variable. It is straightforward, however, to construct constant bit rate Transport Streams with multiple variable rate programs.

2.4.2.3 Buffering

Complete Transport Stream packets containing data from elementary stream n, as indicated by its PID, are passed to the transport buffer for stream n, TB_n. This includes duplicate Transport Stream packets and packets with no payload. Transfer of the i-th byte from the system target decoder input to TB_n is instantaneous, so that the i-th byte enters the buffer for stream n, of size TBS_n, at time t(i).

All bytes that enter the buffer TB_n are removed at the rate Rx_n specified below. Bytes which are part of the PES packet or its contents are delivered to the main buffer B_n for audio elementary streams and system data, and to the multiplexing buffer MB_n for video elementary streams. Other bytes are not, and may be used to control the system. Duplicate Transport Stream packets are not delivered to B_n, MB_n, or B_{sys}.

The buffer TB_n is emptied as follows:

- When there is no data in TB_n, Rx_n is equal to zero.
- Otherwise for video:

$$Rx_n = 1,2 \times R_{max}[\text{profile, level}]$$

where

R_{max}[profile, level] is specified according to the profile and level which can be found in Table 8-13 of ITU-T Rec. H.262 | ISO/IEC 13818-2. This Table specifies the upper bound of the rate of each elementary video stream within a specific profile and level.

Rx_n is equal to $1,2 \times R_{max}$ for ISO/IEC 11172-2 constrained parameter video streams, where R_{max} refers to the maximum bitrate for a Constrained Parameters bitstream in ISO/IEC 11172-2.

For ISO/IEC 13818-7 ADTS audio,

Number of Channels	Rx_n [bit/s]
1-2	2 000 000
3-8	5 529 600
9-12	8 294 400
13-48	33 177 600

Channels: channels that require their own decoder buffer in this elementary stream (i.e. individual channel streams in a single channel element or channel pair element and independently switched coupling channel elements).

For other audio,

$$Rx_n = 2 \times 10^6 \text{ bits per second}$$

For systems data:

$$Rx_n = 1 \times 10^6 \text{ bits per second}$$

Rx_n is measured with respect to the system clock frequency.

Complete Transport Stream packets containing system information, for the program selected for decoding, enter the system transport buffer, TB_{sys}, at the Transport Stream rate. These include Transport Stream packets whose PID values are 0 or 1, and all Transport Stream packets identified via the Program Association Table (see Table 2-25) as having the program_map_PID value for the selected program. Network Information Table (NIT) data as specified by the NIT PID is not transferred to TB_{sys}.

Bytes are removed from TB_{sys} at the rate Rx_{sys} and delivered to B_{sys}. Each byte is transferred instantaneously.

Duplicate Transport Stream packets are not delivered to B_{sys}.

Transport packets which do not enter any TB_n or TB_{sys} are discarded.

The transport buffer size is fixed at 512 bytes.

The elementary stream buffer sizes EBS_1 through EBS_n are defined for video as equal to the vbv_buffer_size as it is carried in the sequence header. Refer to Summary of Constrained Parameters in ISO/IEC 11172-2 and Table 8-14 of ITU-T Rec. H.262 | ISO/IEC 13818-2.

The multiplexing buffer size MBS_1 through MBS_n are defined for video as follows:

For Low and Main level

$$MBS_n = BS_{mux} + BS_{oh} + VBV_{max}[\text{profile, level}] - \text{vbv\_buffer\_size}$$

where BS_{oh}, PES packet overhead buffering is defined as:

$$BS_{oh} = (1/750) \text{ seconds} \times R_{max}[\text{profile, level}]$$

and BS_{mux}, additional multiplex buffering is defined as:

$$BS_{mux} = 0.004 \text{ seconds} \times R_{max}[\text{profile, level}]$$

and where VBV_{max}[profile, level] is defined in Table 8-14 in ITU-T Rec. H.262 | ISO/IEC 13812-2 and R_{max}[profile, level] is defined in Table 8-13 in ITU-T Rec. H.262 | ISO/IEC 13818-2, and vbv buffer size is carried in the sequence header described in 6.2.2 of ITU-T Rec. H.262 | ISO/IEC 13818-2.

For High 1440 and High level

$$MBS_n = BS_{mux} + BS_{oh}$$

where BS_{oh} is defined as:

$$BS_{oh} = (1/750) \text{ seconds} \times R_{max}[\text{profile, level}]$$

and BS_{mux} is defined as:

$$BS_{mux} = 0.004 \text{ seconds} \times R_{max}[\text{profile, level}]$$

and where R_{max}[profile, level] is defined in Table 8-13 in ITU-T Rec. H.262 | ISO/IEC 13818-2.

For Constrained Parameters ISO/IEC 11172-2 bitstreams

$$MBS_n = BS_{mux} + BS_{oh} + vbv\_max - vbv\_buffer\_size$$

where BS_{oh} is defined as:

$$BS_{oh} = (1/750) \text{ seconds} \times R_{max}$$

and BS_{mux} is defined as:

$$BS_{mux} = 0.004 \text{ seconds} \times R_{max}$$

and where R_{max} and vbv_max refer to the maximum bitrate and the maximum vbv_buffer_size for a Constrained Parameters bitstream in ISO/IEC 11172-2 respectively.

A portion BS_{mux} = 4 ms × R_{max}[profile, level] of the MBS_n is allocated for buffering to allow multiplexing. The remainder is available for BS_{oh} and may also be available for initial multiplexing.

NOTE 1 – Buffer occupancy by PES packet overhead is directly bounded in PES streams by the PES-STD which is defined in 2.5.2.4. It is possible, but not necessary, to utilize PES streams to construct Transport Streams.

Buffer BS_n

The main buffer sizes BS_1 through BS_n are defined as follows.

Audio

For ISO/IEC 13818-7 ADTS audio,

Number of Channels	BS_n [bytes]
1-2	3 584
3-8	8 976
9-12	12 804
13-48	51 216

Channels: channels that require their own decoder buffer in this elementary stream (i.e. individual channel streams in a single channel element or channel pair element and independently switched coupling channel elements).

For other audio,

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh} = 3584 \text{ bytes}$$

The size of the access unit decoding buffer BS_{dec}, and the PES packet overhead buffer BS_{oh} are constrained by

$$BS_{dec} + BS_{oh} \leq 2848 \text{ bytes}$$

A portion (736 bytes) of the 3584 byte buffer is allocated for buffering to allow multiplexing. The rest, 2848 bytes, are shared for access unit buffering BS_{dec}, BS_{oh} and additional multiplexing.

Systems

The main buffer B_{sys} for system data is of size BS_{sys} = 1536 bytes.

Video

For video elementary streams, data is transferred from MB_n to EB_n using one of two methods: the leak method or the VBV delay method.

Leak method

The leak method transfers data from MB_n to EB_n using a leak rate R_{bx}. The leak method is used whenever any of the following is true:

- the STD descriptor (refer to 2.6.32) for the elementary stream is not present in the Transport Stream;
- the STD descriptor is present and the leak_valid flag has a value of '1';
- the STD descriptor is present, the leak_valid has a value of '0', and the vbv_delay fields coded in the video stream have the value 0xFFFF; or
- trick mode status is true (refer to 2.4.3.6).

For Low and Main level:

$$Rbx_n = R_{max}[profile, level]$$

For High-1440 and High level:

$$Rbx_n = Min \{1.05 \times R_{es}, R_{max}[profile, level]\}$$

For Constrained Parameters bitstream in ISO/IEC 11172-2:

$$Rbx_n = 1,2 \times R_{max}$$

where R_{max} is the maximum bit rate for a Constrained Parameters bitstream in ISO/IEC 11172-2.

If there is PES packet payload data in MB_n, and buffer EB_n is not full, the PES packet payload is transferred from MB_n to EB_n at a rate equal to R_{bx}. If EB_n is full, data are not removed from MB_n. When a byte of data is transferred from MB_n to EB_n, all PES packet header bytes that are in MB_n and immediately precede that byte, are instantaneously removed and discarded. When there is no PES packet payload data present in MB_n, no data is removed from MB_n. All data that enters MB_n leaves it. All PES packet payload data bytes enter EB_n instantaneously upon leaving MB_n.

Vbv_delay method

The vbv_delay method specifies precisely the time at which each byte of coded video data is transferred from MB_n to EB_n, using the vbv_delay values coded in the video elementary stream. The vbv_delay method is used whenever the STD descriptor (refer to 2.6.32) for this elementary stream is present in the Transport Stream, the leak_valid flag in the descriptor has the value '0', and vbv_delay fields coded in the video stream are not equal to 0xFFFF. If any vbv_delay values in a video sequence are not equal to 0xFFFF, none of the vbv_delay fields in that sequence shall be equal to 0xFFFF (refer to ISO/IEC 11172-2 and ITU-T Rec. H.262 | ISO/IEC 13818-2).

When the vbv_delay method is used, the final byte of the video picture start code for picture j is transferred from MB_n to the EB_n at the time $td_n(j)$ – vbv_delay(j), where $td_n(j)$ is the decoding time of picture j, as defined above, and vbv_delay(j) is the delay time, in seconds, indicated by the vbv_delay field of picture j. The transfer of bytes between the final bytes of successive picture start codes (including the final byte of the second start code), into the buffer EB_n, is at a piecewise constant rate, $R_{bx}(j)$, which is specified for each picture j. Specifically, the rate, $R_{bx}(j)$, of transfer into this buffer is given by:

$$R_{bx}(j) = NB(j) / (vbv\_delay(j) - vbv\_delay(j + 1) + td_n(j + 1) - td_n(j)) \qquad (2\text{-}6)$$

where NB(j) is the number of bytes between the final bytes of the picture start codes (including the final byte of the second start code) of pictures j and j + 1, excluding PES packet header bytes.

NOTE 2 – vbv_delay(j + 1) and $td_n(j + 1)$ may have values that differ from those normally expected for periodic video display if the low_delay flag in the video sequence extension is set to '1'. It may not be possible to determine the correct values by examination of the bit stream.

The $R_{bx}(j)$ derived from equation 2-6 shall be less than or equal to R_{max}[profile, level] for elementary streams of stream type 0x02 (refer to Table 2-29), where R_{max}[profile, level] is defined in ITU-T Rec. H.262 | ISO/IEC 13818-2, and shall be less than or equal to the maximum bit rate allowed for constrained parameter video elementary streams of stream type 0x01, refer to ISO/IEC 11172-2.

When a byte of data is transferred from MB_n to EB_n, all PES packet header bytes that are in MB_n and immediately precede that byte are instantaneously removed and discarded. All data that enters MB_n leaves it. All PES packet payload data bytes enter EB_n instantaneously upon leaving MB_n.

Removal of access units

For each elementary stream buffer EB_n and main buffer B_n all data for the access unit that has been in the buffer longest, $A_n(j)$, and any stuffing bytes that immediately precede it that are present in the buffer at the time $td_n(j)$ are removed instantaneously at time $td_n(j)$. The decoding time $td_n(j)$ is specified in the DTS or PTS fields (refer to 2.4.3.6). Decoding times $td_n(j + 1)$, $td_n(j + 2)$, ... of access units without encoded DTS or PTS fields which directly follow access unit j may be derived from information in the elementary stream. Refer to Annex C of ITU-T Rec. H.262 | ISO/IEC 13818-2, ISO/IEC 13818-3, or ISO/IEC 11172. Also refer to 2.7.5. In the case of audio, all PES packet headers that are stored immediately before the access unit or that are embedded within the data of the access unit are removed simultaneously with the removal of the access unit. As the access unit is removed it is instantaneously decoded to a presentation unit.

System data

In the case of system data, data is removed from the main buffer B_{sys} at a rate of R_{sys} whenever there is at least 1 byte available in buffer B_{sys}.

$$R_{sys} = \max(80\,000\ bits/s,\ transport\_rate(i) \times 8\ bits/byte / 500) \qquad (2\text{-}7)$$

NOTE 3 – The intention of increasing R_{sys} in the case of high transport rates is to allow an increased data rate for the Program Specific Information.

Low delay

When the low_delay flag in the video sequence extension is set to '1' (see 6.2.2.3 of ITU-T Rec. H.262 | ISO/IEC 13818-2) the EB_n buffer may underflow. In this case, when the T-STD elementary stream buffer EB_n is examined at the time specified by $td_n(j)$, the complete data for the access unit may not be present in the buffer EB_n. When this case arises, the buffer shall be re-examined at intervals of two field-periods until the data for the complete access unit is present in the buffer. At this time the entire access unit shall be removed from buffer EB_n instantaneously. Overflow of buffer EB_n shall not occur.

When the low_delay_mode flag is set to '1', EB_n underflow is allowed to occur continuously without limit. The T-STD decoder shall remove access unit data from buffer EB_n at the earliest time consistent with the paragraph above and any DTS or PTS values encoded in the bit stream. Note that the decoder may be unable to re-establish correct decoding and display times as indicated by DTS and PTS until the EB_n buffer underflow situation ceases and a PTS or DTS is found in the bit stream.

Trick mode

When the DSM_trick_mode flag (2.4.3.6) is set to '1' in the PES Packet header of a packet containing the start of a B-type video access unit and the trick_mode_control field is set to '001' (slow motion) or '010' (freeze frame), or '100' (slow reverse) the B-picture access unit is not removed from the video data buffer EB_n until the last time of possibly multiple times that any field of the picture is decoded and presented. Repetition of the presentation of fields and pictures is defined in 2.4.3.8 under slow motion, slow reverse, and field_id_cntrl. The access unit is removed instantaneously from EB_n at the indicated time, which is dependent on the value of rep_cntrl.

When the DSM_trick_mode flag is set to '1' in the PES packet header of a packet containing the first byte of a picture start code, trick_mode status becomes true when that picture start code in the PES packet is removed from buffer EB_n. Trick mode status remains true until a PES packet header is received by the T-STD in which the DSM_trick_mode flag is set to '0' and the first byte of the picture start code after that PES packet header is removed from buffer EB_n. When trick mode status is true, the buffer EB_n may underflow. All other constraints from normal streams are retained when trick mode status is true.

2.4.2.4 Decoding

Elementary streams buffered in B_1 through B_n and EB_1 through EB_n are decoded instantaneously by decoders D_1 through D_n and may be delayed in re-order buffers O_1 through O_n before being presented at the output of the T-STD. Re-order buffers are used only in the case of a video elementary stream when some access units are not carried in presentation order. These access units will need to be re-ordered before presentation. In particular, if $P_n(k)$ is an I-picture or a P-picture carried before one or more B-pictures, then it must be delayed in the re-order buffer, O_n, of the T-STD before being presented. Any picture previously stored in O_n is presented before the current picture can be stored. $P_n(k)$ should be delayed until the next I-picture or P-picture is decoded. While it is stored in the re-order buffer, the subsequent B-pictures are decoded and presented.

The time at which a presentation unit $P_n(k)$ is presented is $tp_n(k)$. For presentation units that do not require re-ordering delay, $tp_n(k)$ is equal to $td_n(j)$ since the access units are decoded instantaneously; this is the case, for example, for B-frames. For presentation units that are delayed, $tp_n(k)$ and $td_n(j)$ differ by the time that $P_n(k)$ is delayed in the re-order buffer, which is a multiple of the nominal picture period. Care should be taken to use adequate re-ordering delay from the beginning of video elementary streams to meet the requirements of the entire stream. For example, a stream which initially has only I- and P-pictures but later includes B-pictures should include re-ordering delay starting at the beginning of the stream.

ITU-T Rec. H.262 | ISO/IEC 13818-2 explains re-ordering of video pictures in greater detail.

2.4.2.5 Presentation

The function of a decoding system is to reconstruct presentation units from compressed data and to present them in a synchronized sequence at the correct presentation times. Although real audio and visual presentation devices generally have finite and different delays and may have additional delays imposed by post-processing or output functions, the system target decoder models these delays as zero.

In the T-STD in Figure 2-1 the display of a video presentation unit (a picture) occurs instantaneously at its presentation time, $tp_n(k)$.

In the T-STD the output of an audio presentation unit starts at its presentation time, $tp_n(k)$, when the decoder instantaneously presents the first sample. Subsequent samples in the presentation unit are presented in sequence at the audio sampling rate.

2.4.2.6 Buffer management

Transport Streams shall be constructed so that conditions defined in this subclause are satisfied. This subclause makes use of the notation defined for the System Target Decoder.

TB_n and TB_{sys} shall not overflow. TB_n and TB_{sys} shall empty at least once every second. B_n shall not overflow nor underflow. B_{sys} shall not overflow.

EB_n shall not underflow except when the low delay flag in the video sequence extension is set to '1' (refer to 6.2.2.3 in ITU-T Rec. H.262 | ISO/IEC 13818-2) or trick_mode status is true.

When the leak method for specifying transfers is in effect, MB_n shall not overflow, and shall empty at least once every second. EB_n shall not overflow.

When the vbv_delay method for specifying transfers is in effect, MB_n shall not overflow nor underflow, and EB_n shall not overflow.

The delay of any data through the System Target Decoder buffers shall be less than or equal to one second except for still picture video data and ISO/IEC 14496 streams. Specifically: $td_n(j) - t(i) \leq 1$ second for all j, and all bytes i in access unit $A_n(j)$.

For still picture video data, the delay is constrained by $td_n(j) - t(i) \leq 60$ seconds for all j, and all bytes i in access unit $A_n(j)$.

For ISO/IEC 14496 streams, the delay is constrained by $td_n(j)-t(i) \leq 10$ second for all j, and all bytes i in access unit $A_n(j)$.

Definition of overflow and underflow

Let $F_n(t)$ be the instantaneous fullness of T-STD buffer B_n.

$F_n(t) = 0$ instantaneously before $t = t(0)$

Overflow does not occur if

$$F_n(t) \leq BS_n$$

for all t and n.

Underflow does not occur if

$$0 \leq F_n(t)$$

for all t and n.

2.4.2.7 T-STD extensions for carriage of ISO/IEC 14496 data

For decoding of ISO/IEC 14496 data carried in a Transport Stream the T-STD model is extended. T-STD parameters for decoding of individual ISO/IEC 14496 elementary streams are defined in subclause 2.11.2, while subclause 2.11.3 defines T-STD extensions and parameters for decoding of ISO/IEC 14496 scenes and associated streams.

2.4.3 Specification of the Transport Stream syntax and semantics

The following syntax describes a stream of bytes. Transport Stream packets shall be 188 bytes long.

2.4.3.1 Transport Stream

See Table 2-1.

Table 2-1 – Transport Stream

Syntax	No. of bits	Mnemonic

MPEG_transport_stream() {
do {
transport_packet()
} while (nextbits() == sync_byte)
}		

2.4.3.2 Transport Stream packet layer

See Table 2-2.

Table 2-2 – Transport packet of this Recommendation | International Standard

Syntax	No. of bits	Mnemonic
transport_packet(){		
sync_byte	**8**	**bslbf**
transport_error_indicator	**1**	**bslbf**
payload_unit_start_indicator	**1**	**bslbf**
transport_priority	**1**	**bslbf**
PID	**13**	**uimsbf**
transport_scrambling_control	**2**	**bslbf**
adaptation_field_control	**2**	**bslbf**
continuity_counter	**4**	**uimsbf**
if(adaptation_field_control == '10' || adaptation_field_control == '11'){		
adaptation_field()		
}		
if(adaptation_field_control == '01' || adaptation_field_control == '11') {		
for (i = 0; i < N; i++){		
data_byte	**8**	**bslbf**
}		
}		
}		

2.4.3.3 Semantic definition of fields in Transport Stream packet layer

sync_byte – The sync_byte is a fixed 8-bit field whose value is '0100 0111' (0x47). Sync_byte emulation in the choice of values for other regularly occurring fields, such as PID, should be avoided.

transport_error_indicator – The transport_error_indicator is a 1-bit flag. When set to '1' it indicates that at least 1 uncorrectable bit error exists in the associated Transport Stream packet. This bit may be set to '1' by entities external to the transport layer. When set to '1' this bit shall not be reset to '0' unless the bit value(s) in error have been corrected.

payload_unit_start_indicator – The payload_unit_start_indicator is a 1-bit flag which has normative meaning for Transport Stream packets that carry PES packets (refer to 2.4.3.6) or PSI data (refer to 2.4.4).

When the payload of the Transport Stream packet contains PES packet data, the payload_unit_start_indicator has the following significance: a '1' indicates that the payload of this Transport Stream packet will commence with the first byte of a PES packet and a '0' indicates no PES packet shall start in this Transport Stream packet. If the payload_unit_start_indicator is set to '1', then one and only one PES packet starts in this Transport Stream packet. This also applies to private streams of stream_type 6 (refer to Table 2-29).

When the payload of the Transport Stream packet contains PSI data, the payload_unit_start_indicator has the following significance: if the Transport Stream packet carries the first byte of a PSI section, the payload_unit_start_indicator value shall be '1', indicating that the first byte of the payload of this Transport Stream packet carries the pointer_field. If the Transport Stream packet does not carry the first byte of a PSI section, the payload_unit_start_indicator value shall be '0', indicating that there is no pointer_field in the payload. Refer to 2.4.4.1 and 2.4.4.2. This also applies to private streams of stream_type 5 (refer to Table 2-29).

For null packets the payload_unit_start_indicator shall be set to '0'.

The meaning of this bit for Transport Stream packets carrying only private data is not defined in this Specification.

transport_priority – The transport_priority is a 1-bit indicator. When set to '1' it indicates that the associated packet is of greater priority than other packets having the same PID which do not have the bit set to '1'. The transport mechanism can use this to prioritize its data within an elementary stream. Depending on the application the transport_priority field may be coded regardless of the PID or within one PID only. This field may be changed by channel specific encoders or decoders.

PID – The PID is a 13-bit field, indicating the type of the data stored in the packet payload. PID value 0x0000 is reserved for the Program Association Table (see Table 2-25). PID value 0x0001 is reserved for the Conditional Access Table (see Table 2-27). PID values 0x0002 – 0x000F are reserved. PID value 0x1FFF is reserved for null packets (see Table 2-3).

Table 2-3 – PID table

Value	Description
0x0000	Program Association Table
0x0001	Conditional Access Table
0x0002	Transport Stream Description Table
0x0003-0x000F	Reserved
0x00010 … 0x1FFE	May be assigned as network_PID, Program_map_PID, elementary_PID, or for other purposes
0x1FFF	Null packet
NOTE – The transport packets with PID values 0x0000, 0x0001, and 0x0010-0x1FFE are allowed to carry a PCR.	

transport_scrambling_control – This 2-bit field indicates the scrambling mode of the Transport Stream packet payload. The Transport Stream packet header, and the adaptation field when present, shall not be scrambled. In the case of a null packet the value of the transport_scrambling_control field shall be set to '00' (see Table 2-4).

Table 2-4 – Scrambling control values

Value	Description
00	Not scrambled
01	User-defined
10	User-defined
11	User-defined

adaptation_field_control – This 2-bit field indicates whether this Transport Stream packet header is followed by an adaptation field and/or payload (see Table 2-5).

Table 2-5 – Adaptation field control values

Value	Description
00	Reserved for future use by ISO/IEC
01	No adaptation_field, payload only
10	Adaptation_field only, no payload
11	Adaptation_field followed by payload

ITU-T Rec. H.222.0 | ISO/IEC 13818-1 decoders shall discard Transport Stream packets with the adaptation_field_control field set to a value of '00'. In the case of a null packet the value of the adaptation_field_control shall be set to '01'.

continuity_counter – The continuity_counter is a 4-bit field incrementing with each Transport Stream packet with the same PID. The continuity_counter wraps around to 0 after its maximum value. The continuity_counter shall not be incremented when the adaptation_field_control of the packet equals '00' or '10'.

In Transport Streams, duplicate packets may be sent as two, and only two, consecutive Transport Stream packets of the same PID. The duplicate packets shall have the same continuity_counter value as the original packet and the adaptation_field_control field shall be equal to '01' or '11'. In duplicate packets each byte of the original packet shall be duplicated, with the exception that in the program clock reference fields, if present, a valid value shall be encoded.

The continuity_counter in a particular Transport Stream packet is continuous when it differs by a positive value of one from the continuity_counter value in the previous Transport Stream packet of the same PID, or when either of the non-incrementing conditions (adaptation_field_control set to '00' or '10', or duplicate packets as described above) are met. The continuity counter may be discontinuous when the discontinuity_indicator is set to '1' (refer to 2.4.3.4). In the case of a null packet the value of the continuity_counter is undefined.

data_byte – Data bytes shall be contiguous bytes of data from the PES packets (refer to 2.4.3.6), PSI sections (refer to 2.4.4), packet stuffing bytes after PSI sections, or private data not in these structures as indicated by the PID. In the case of null packets with PID value 0x1FFF, data_bytes may be assigned any value. The number of data_bytes, N, is specified by 184 minus the number of bytes in the adaptation_field(), as described in 2.4.3.4 below.

2.4.3.4 Adaptation field

See Table 2-6.

2.4.3.5 Semantic definition of fields in adaptation field

adaptation_field_length – The adaptation_field_length is an 8-bit field specifying the number of bytes in the adaptation_field immediately following the adaptation_field_length. The value 0 is for inserting a single stuffing byte in a Transport Stream packet. When the adaptation_field_control value is '11', the value of the adaptation_field_length shall be in the range 0 to 182. When the adaptation_field_control value is '10', the value of the adaptation_field_length shall be 183. For Transport Stream packets carrying PES packets, stuffing is needed when there is insufficient PES packet data to completely fill the Transport Stream packet payload bytes. Stuffing is accomplished by defining an adaptation field longer than the sum of the lengths of the data elements in it, so that the payload bytes remaining after the adaptation field exactly accommodates the available PES packet data. The extra space in the adaptation field is filled with stuffing bytes.

This is the only method of stuffing allowed for Transport Stream packets carrying PES packets. For Transport Stream packets carrying PSI, an alternative stuffing method is described in 2.4.4.

discontinuity_indicator – This is a 1-bit field which when set to '1' indicates that the discontinuity state is true for the current Transport Stream packet. When the discontinuity_indicator is set to '0' or is not present, the discontinuity state is false. The discontinuity indicator is used to indicate two types of discontinuities, system time-base discontinuities and continuity_counter discontinuities.

A system time-base discontinuity is indicated by the use of the discontinuity_indicator in Transport Stream packets of a PID designated as a PCR_PID (refer to 2.4.4.9). When the discontinuity state is true for a Transport Stream packet of a PID designated as a PCR_PID, the next PCR in a Transport Stream packet with that same PID represents a sample of a new system time clock for the associated program. The system time-base discontinuity point is defined to be the instant in time when the first byte of a packet containing a PCR of a new system time-base arrives at the input of the T-STD. The discontinuity_indicator shall be set to '1' in the packet in which the system time-base discontinuity occurs. The discontinuity_indicator bit may also be set to '1' in Transport Stream packets of the same PCR_PID prior to the packet which contains the new system time-base PCR. In this case, once the discontinuity_indicator has been set to '1', it shall continue to be set to '1' in all Transport Stream packets of the same PCR_PID up to and including the Transport Stream packet which contains the first PCR of the new system time-base. After the occurrence of a system time-base discontinuity, no fewer than two PCRs for the new system time-base shall be received before another system time-base discontinuity can occur. Further, except when trick mode status is true, data from no more than two system time-bases shall be present in the set of T-STD buffers for one program at any time.

Prior to the occurrence of a system time-base discontinuity, the first byte of a Transport Stream packet which contains a PTS or DTS which refers to the new system time-base shall not arrive at the input of the T-STD. After the occurrence of a system time-base discontinuity, the first byte of a Transport Stream packet which contains a PTS or DTS which refers to the previous system time-base shall not arrive at the input of the T-STD.

A continuity_counter discontinuity is indicated by the use of the discontinuity_indicator in any Transport Stream packet. When the discontinuity state is true in any Transport Stream packet of a PID not designated as a PCR_PID, the continuity_counter in that packet may be discontinuous with respect to the previous Transport Stream packet of the same PID. When the discontinuity state is true in a Transport Stream packet of a PID that is designated as a PCR_PID, the continuity_counter may only be discontinuous in the packet in which a system time-base discontinuity occurs. A continuity counter discontinuity point occurs when the discontinuity state is true in a Transport Stream packet and the continuity_counter in the same packet is discontinuous with respect to the previous Transport Stream packet of the same PID. A continuity counter discontinuity point shall occur at most one time from the initiation of the discontinuity state until the conclusion of the discontinuity state. Furthermore, for all PIDs that are not designated as PCR_PIDs, when the discontinuity_indicator is set to '1' in a packet of a specific PID, the discontinuity_indicator may be set to '1' in the next Transport Stream packet of that same PID, but shall not be set to '1' in three consecutive Transport Stream packet of that same PID.

For the purpose of this clause, an elementary stream access point is defined as follows:

- Video – The first byte of a video sequence header.
- Audio – The first byte of an audio frame.

After a continuity counter discontinuity in a Transport packet which is designated as containing elementary stream data, the first byte of elementary stream data in a Transport Stream packet of the same PID shall be the first byte of an elementary stream access point or in the case of video, the first byte of an elementary stream access point or a sequence_end_code followed by an access point. Each Transport Stream packet which contains elementary stream data with a PID not designated as a PCR_PID, and in which a continuity counter discontinuity point occurs, and in which a PTS or DTS occurs, shall arrive at the input of the T-STD after the system time-base discontinuity for the associated program occurs. In the case where the discontinuity state is true, if two consecutive Transport Stream packets of the same PID occur which have the same continuity_counter value and have adaptation_field_control values set to '01' or '11', the second packet may be discarded. A Transport Stream shall not be constructed in such a way that discarding such a packet will cause the loss of PES packet payload data or PSI data.

Table 2-6 – Transport Stream adaptation field

Syntax	No. of bits	Mnemonic
adaptation_field() {		
adaptation_field_length	**8**	**uimsbf**
if (adaptation_field_length > 0) {		
discontinuity_indicator	**1**	**bslbf**
random_access_indicator	**1**	**bslbf**
elementary_stream_priority_indicator	**1**	**bslbf**
PCR_flag	**1**	**bslbf**
OPCR_flag	**1**	**bslbf**
splicing_point_flag	**1**	**bslbf**
transport_private_data_flag	**1**	**bslbf**
adaptation_field_extension_flag	**1**	**bslbf**
if (PCR_flag == '1') {		
program_clock_reference_base	**33**	**uimsbf**
reserved	**6**	**bslbf**
program_clock_reference_extension	**9**	**uimsbf**
}		
if (OPCR_flag == '1') {		
original_program_clock_reference_base	**33**	**uimsbf**
reserved	**6**	**bslbf**
original_program_clock_reference_extension	**9**	**uimsbf**
}		
if (splicing_point_flag == '1') {		
splice_countdown	**8**	**tcimsbf**
}		
if (transport_private_data_flag == '1') {		
transport_private_data_length	**8**	**uimsbf**
for (i = 0; i < transport_private_data_length; i++) {		
private_data_byte	**8**	**bslbf**
}		
}		
if (adaptation_field_extension_flag == '1') {		
adaptation_field_extension_length	**8**	**uimsbf**
ltw_flag	**1**	**bslbf**
piecewise_rate_flag	**1**	**bslbf**
seamless_splice_flag	**1**	**bslbf**
reserved	**5**	**bslbf**
if (ltw_flag == '1') {		
ltw_valid_flag	**1**	**bslbf**
ltw_offset	**15**	**uimsbf**
}		
if (piecewise_rate_flag == '1') {		
reserved	**2**	**bslbf**
piecewise_rate	**22**	**uimsbf**
}		
if (seamless_splice_flag == '1') {		
splice_type	**4**	**bslbf**
DTS_next_AU[32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
DTS_next_AU[29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
DTS_next_AU[14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
for (i = 0; i < N; i++) {		
reserved	**8**	**bslbf**
}		
}		
for (i = 0; i < N; i++) {		
stuffing_byte	**8**	**bslbf**
}		
}		
}		

After the occurrence of a discontinuity_indicator set to '1' in a Transport Stream packet which contains PSI information, a single discontinuity in the version_number of PSI sections may occur. At the occurrence of such a discontinuity, a version of the TS_program_map_sections of the appropriate program shall be sent with section_length == 13 and the current_next_indicator == 1, such that there are no program_descriptors and no elementary streams described. This shall then be followed by a version of the TS_program_map_section for each affected program with the version_number incremented by one and the current_next_indicator == 1, containing a complete program definition. This indicates a version change in PSI data.

random_access_indicator – The random_access_indicator is a 1-bit field that indicates that the current Transport Stream packet, and possibly subsequent Transport Stream packets with the same PID, contain some information to aid random access at this point. Specifically, when the bit is set to '1', the next PES packet to start in the payload of Transport Stream packets with the current PID shall contain the first byte of a video sequence header if the PES stream type (refer to Table 2-29) is 1 or 2, or shall contain the first byte of an audio frame if the PES stream type is 3 or 4. In addition, in the case of video, a presentation timestamp shall be present in the PES packet containing the first picture following the sequence header. In the case of audio, the presentation timestamp shall be present in the PES packet containing the first byte of the audio frame. In the PCR_PID the random_access_indicator may only be set to '1' in Transport Stream packet containing the PCR fields.

elementary_stream_priority_indicator – The elementary_stream_priority_indicator is a 1-bit field. It indicates, among packets with the same PID, the priority of the elementary stream data carried within the payload of this Transport Stream packet. A '1' indicates that the payload has a higher priority than the payloads of other Transport Stream packets. In the case of video, this field may be set to '1' only if the payload contains one or more bytes from an intra-coded slice. A value of '0' indicates that the payload has the same priority as all other packets which do not have this bit set to '1'.

PCR_flag – The PCR_flag is a 1-bit flag. A value of '1' indicates that the adaptation_field contains a PCR field coded in two parts. A value of '0' indicates that the adaptation field does not contain any PCR field.

OPCR_flag – The OPCR_flag is a 1-bit flag. A value of '1' indicates that the adaptation_field contains an OPCR field coded in two parts. A value of '0' indicates that the adaptation field does not contain any OPCR field.

splicing_point_flag – The splicing_point_flag is a 1-bit flag. When set to '1', it indicates that a splice_countdown field shall be present in the associated adaptation field, specifying the occurrence of a splicing point. A value of '0' indicates that a splice_countdown field is not present in the adaptation field.

transport_private_data_flag – The transport_private_data_flag is a 1-bit flag. A value of '1' indicates that the adaptation field contains one or more private_data bytes. A value of '0' indicates the adaptation field does not contain any private_data bytes.

adaptation_field_extension_flag – The adaptation_field_extension_flag is a 1-bit field which when set to '1' indicates the presence of an adaptation field extension. A value of '0' indicates that an adaptation field extension is not present in the adaptation field.

program_clock_reference_base; program_clock_reference_extension – The program_clock_reference (PCR) is a 42-bit field coded in two parts. The first part, program_clock_reference_base, is a 33-bit field whose value is given by PCR_base(i), as given in equation 2-2. The second part, program_clock_reference_extension, is a 9-bit field whose value is given by PCR_ext(i), as given in equation 2-3. The PCR indicates the intended time of arrival of the byte containing the last bit of the program_clock_reference_base at the input of the system target decoder.

original_program_clock_reference_base; original_program_clock_reference_extension – The optional original program reference (OPCR) is a 42-bit field coded in two parts. These two parts, the base and the extension, are coded identically to the two corresponding parts of the PCR field. The presence of the OPCR is indicated by the OPCR_flag. The OPCR field shall be coded only in Transport Stream packets in which the PCR field is present. OPCRs are permitted in both single program and multiple program Transport Streams.

OPCR assists in the reconstruction of a single program Transport Stream from another Transport Stream. When reconstructing the original single program Transport Stream, the OPCR may be copied to the PCR field. The resulting PCR value is valid only if the original single program Transport Stream is reconstructed exactly in its entirety. This would include at least any PSI and private data packets which were present in the original Transport Stream and would possibly require other private arrangements. It also means that the OPCR must be an identical copy of its associated PCR in the original single program Transport Stream.

The OPCR is expressed as follows:

$$OPCR(i) = OPCR\_base(i) \times 300 + OPCR\_ext(i) \quad (2\text{-}8)$$

where

$$OPCR\_base(i) = ((system\_clock\_frequency \times t(i)) \; DIV \; 300) \; \% \; 2^{33} \quad (2\text{-}9)$$

$$OPCR\_ext(i) = ((system\_clock\_frequency \times t(i)) \; DIV \; 1) \; \% \; 300 \quad (2\text{-}10)$$

The OPCR field is ignored by the decoder. The OPCR field shall not be modified by any multiplexor or decoder.

splice_countdown – The splice_countdown is an 8-bit field, representing a value which may be positive or negative. A positive value specifies the remaining number of Transport Stream packets, of the same PID, following the associated Transport Stream packet until a splicing point is reached. Duplicate Transport Stream packets and Transport Stream packets which only contain adaptation fields are excluded. The splicing point is located immediately after the last byte of the Transport Stream packet in which the associated splice_countdown field reaches zero. In the Transport Stream packet where the splice_countdown reaches zero, the last data byte of the Transport Stream packet payload shall be the last byte of a coded audio frame or a coded picture. In the case of video, the corresponding access unit may or may not be terminated by a sequence_end_code. Transport Stream packets with the same PID, which follow, may contain data from a different elementary stream of the same type.

The payload of the next Transport Stream packet of the same PID (duplicate packets and packets without payload being excluded) shall commence with the first byte of a PES packet. In the case of audio, the PES packet payload shall commence with an access point. In the case of video, the PES packet payload shall commence with an access point, or with a sequence_end_code, followed by an access point. Thus, the previous coded audio frame or coded picture aligns with the packet boundary, or is padded to make this so. Subsequent to the splicing point, the countdown field may also be present. When the splice_countdown is a negative number whose value is minus n (–n), it indicates that the associated Transport Stream packet is the n-th packet following the splicing point (duplicate packets and packets without payload being excluded).

For the purposes of this subclause, an access point is defined as follows:

- Video – The first byte of a video_sequence_header.
- Audio – The first byte of an audio frame.

transport_private_data_length – The transport_private_data_length is an 8-bit field specifying the number of private_data bytes immediately following the transport private_data_length field. The number of private_data bytes shall not be such that private data extends beyond the adaptation field.

private_data_byte – The private_data_byte is an 8-bit field that shall not be specified by ITU-T | ISO/IEC.

adaptation_field_extension_length – The adaptation_field_extension_length is an 8-bit field. It indicates the number of bytes of the extended adaptation field data immediately following this field, including reserved bytes if present.

ltw_flag (legal time window_flag) – This is a 1-bit field which when set to '1' indicates the presence of the ltw_offset field.

piecewise_rate_flag – This is a 1-bit field which when set to '1' indicates the presence of the piecewise_rate field.

seamless_splice_flag – This is a 1-bit flag which when set to '1' indicates that the splice_type and DTS_next_AU fields are present. A value of '0' indicates that neither splice_type nor DTS_next_AU fields are present. This field shall not be set to '1' in Transport Stream packets in which the splicing_point_flag is not set to '1'. Once it is set to '1' in a Transport Stream packet in which the splice_countdown is positive, it shall be set to '1' in all the subsequent Transport Stream packets of the same PID that have the splicing_point_flag set to '1', until the packet in which the splice_countdown reaches zero (including this packet). When this flag is set, if the elementary stream carried in this PID is an audio stream, the splice_type field shall be set to '0000'. If the elementary stream carried in this PID is a video stream, it shall fulfil the constraints indicated by the splice_type value.

ltw_valid_flag (legal time window_valid_flag) – This is a 1-bit field which when set to '1' indicates that the value of the ltw_offset shall be valid. A value of '0' indicates that the value in the ltw_offset field is undefined.

ltw_offset (legal time window offset) – This is a 15-bit field, the value of which is defined only if the ltw_valid flag has a value of '1'. When defined, the legal time window offset is in units of $(300/f_s)$ seconds, where f_s is the system clock frequency of the program that this PID belongs to, and fulfils:

$$\text{offset} = t_1(i) - t(i)$$

$$\text{ltw\_offset} = \text{offset}//1$$

where i is the index of the first byte of this Transport Stream packet, offset is the value encoded in this field, t(i) is the arrival time of byte i in the T-STD, and $t_1(i)$ is the upper bound in time of a time interval called the Legal Time Window which is associated with this Transport Stream packet.

The Legal Time Window has the property that if this Transport Stream is delivered to a T-STD starting at time $t_1(i)$, i.e. at the end of its Legal Time Window, and all other Transport Stream packets of the same program are delivered at the end of their Legal Time Windows, then

- For video – The MB_n buffer for this PID in the T-STD shall contain less than 184 bytes of elementary stream data at the time the first byte of the payload of this Transport Stream packet enters it, and no buffer violations in the T-STD shall occur.
- For audio – The B_n buffer for this PID in the T-STD shall contain less than $BS_{dec} + 1$ bytes of elementary stream data at the time the first byte of this Transport Stream packet enters it, and no buffer violations in the T-STD shall occur.

Depending on factors including the size of the buffer MB_n and the rate of data transfer between MB_n and EB_n, it is possible to determine another time $t_0(i)$, such that if this packet is delivered anywhere in the interval $[t_0(i), t_1(i)]$, no T-STD buffer violations will occur. This time interval is called the Legal Time Window. The value of t_0 is not defined in this Recommendation | International Standard.

The information in this field is intended for devices such as remultiplexers which may need this information in order to reconstruct the state of the buffers MB_n.

piecewise_rate – The meaning of this 22-bit field is only defined when both the ltw_flag and the ltw_valid_flag are set to '1'. When defined, it is a positive integer specifying a hypothetical bitrate R which is used to define the end times of the Legal Time Windows of Transport Stream packets of the same PID that follow this packet but do not include the legal_time_window_offset field.

Assume that the first byte of this Transport Stream packet and the N following Transport Stream packets of the same PID have indices A_i, A_{i+1}, ..., A_{i+N}, respectively, and that the N latter packets do not have a value encoded in the field legal_time_window_offset. Then the values $t_1(A_{i+j})$ shall be determined by:

$$t_1(A_{i+j}) = t_1(A_i) + j \times 188 \times \text{8-bits/byte} / R$$

where j goes from 1 to N.

All packets between this packet and the next packet of the same PID to include a legal_time_window_offset field shall be treated as if they had the value:

$$\text{offset} = t_1(A_i) - t(A_i)$$

corresponding to the value $t_1(.)$ as computed by the formula above encoded in the legal_time_window_offset field. t(j) is the arrival time of byte j in the T-STD.

The meaning of this field is not defined when it is present in a Transport Stream packet with no legal_time_window_offset field.

splice_type – This is a 4-bit field. From the first occurrence of this field onwards, it shall have the same value in all the subsequent Transport Stream packets of the same PID in which it is present, until the packet in which the splice_countdown reaches zero (including this packet). If the elementary stream carried in that PID is an audio stream, this field shall have the value '0000'. If the elementary stream carried in that PID is a video stream, this field indicates the conditions that shall be respected by this elementary stream for splicing purposes. These conditions are defined as a function of profile, level and splice_type in Table 2-7 through Table 2-16.

In these tables, a value for 'splice_decoding_delay' and 'max_splice_rate' means that the following conditions shall be satisfied by the video elementary stream:

1) The last byte of the coded picture ending in the Transport Stream packet in which the splice_countdown reaches zero shall remain in the VBV buffer of the VBV model for an amount of time equal to (splice_decoding_delay $t_{n+1} - t_n$), where for the purpose of this subclause:

   - n is the index of the coded picture ending in the Transport Stream packet in which the splice_countdown reaches zero, i.e. the coded picture referred to above.
   - t_n is defined in C.3.1 of ITU-T Rec. H.262 | ISO/IEC 13818-2.
   - $(t_{n+1} - t_n)$ is defined in C.9 through C.12 of ITU-T Rec. H.262 | ISO/IEC 13818-2.

     NOTE – t_n is the time when coded picture n is removed from the VBV buffer, and $(t_{n+1} - t_n)$ is the duration for which picture n is presented.

2) The VBV buffer of the VBV model shall not overflow if its input is switched at the splicing point to a stream of a constant rate equal to 'max_splice_rate' for an amount of time equal to 'splice_decoding_delay'.

Table 2-7 – Splice parameters Table 1

Simple Profile Main Level, Main Profile Main Level, SNR Profile Main Level (both layers), Spatial Profile High-1440 Level (base layer), High Profile Main Level (middle + base layers), Multi-view Profile Main Level (base layer) Video

splice_type	Conditions
0000	splice_decoding_delay = 120 ms; max_splice_rate = 15.0×10^6 bit/s
0001	splice_decoding_delay = 150 ms; max_splice_rate = 12.0×10^6 bit/s
0010	splice_decoding_delay = 225 ms; max_splice_rate = 8.0×10^6 bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = 7.2×10^6 bit/s
0100-1011	Reserved
1100-1111	User-defined

Table 2-8 – Splice parameters Table 2

Main Profile Low Level, SNR Profile Low Level (both layers), High Profile Main Level (base layer), Multi-view Profile Low Level (base layer) Video

splice_type	Conditions
0000	splice_decoding_delay = 115 ms; max_splice_rate = 4.0×10^6 bit/s
0001	splice_decoding_delay = 155 ms; max_splice_rate = 3.0×10^6 bit/s
0010	splice_decoding_delay = 230 ms; max_splice_rate = 2.0×10^6 bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = 1.8×10^6 bit/s
0100-1011	Reserved
1100-1111	User-defined

Table 2-9 – Splice parameters Table 3

Main Profile High-1440 Level, Spatial Profile High-1440 Level (all layers), High Profile High-1440 Level (middle + base layers), Multi-view Profile High-1440 Level (base layer) Video

splice_type	Conditions
0000	splice_decoding_delay = 120 ms; max_splice_rate = 60.0×10^6 bit/s
0001	splice_decoding_delay = 160 ms; max_splice_rate = 45.0×10^6 bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = 30.0×10^6 bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = 28.5×10^6 bit/s
0100-1011	Reserved
1100-1111	User-defined

Table 2-10 – Splice parameters Table 4

Main Profile High Level, High Profile High-1440 Level (all layers), High Profile High Level (middle + base layers), Multi-view Profile High Level (base layer) Video

splice_type	Conditions
0000	splice_decoding_delay = 120 ms; max_splice_rate = 80.0×10^6 bit/s
0001	splice_decoding_delay = 160 ms; max_splice_rate = 60.0×10^6 bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = 40.0×10^6 bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = 38.0×10^6 bit/s
0100-1011	Reserved
1100-1111	User-defined

Table 2-11 – Splice parameters Table 5

SNR Profile Low Level (base layer) Video

splice_type	Conditions
0000	splice_decoding_delay = 115 ms; max_splice_rate = 3.0×10^6 bit/s
0001	splice_decoding_delay = 175 ms; max_splice_rate = 2.0×10^6 bit/s
0010	splice_decoding_delay = 250 ms; max_splice_rate = 1.4×10^6 bit/s
0011-1011	Reserved
1100-1111	User-defined

Table 2-12 – Splice parameters Table 6

SNR Profile Main Level (base layer) Video

splice_type	Conditions
0000	splice_decoding_delay = 115 ms; max_splice_rate = 10.0×10^6 bit/s
0001	splice_decoding_delay = 145 ms; max_splice_rate = 8.0×10^6 bit/s
0010	splice_decoding_delay = 235 ms; max_splice_rate = 5.0×10^6 bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = 4.7×10^6 bit/s
0100-1011	Reserved
1100-1111	User-defined

Table 2-13 – Splice parameters Table 7

Spatial Profile High-1440 Level (middle + base layers) Video

splice_type	Conditions
0000	splice_decoding_delay = 120 ms; max_splice_rate = 40.0×10^6 bit/s
0001	splice_decoding_delay = 160 ms; max_splice_rate = 30.0×10^6 bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = 20.0×10^6 bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = 19.0×10^6 bit/s
0100-1011	Reserved
1100-1111	User-defined

Table 2-14 – Splice parameters Table 8

High Profile Main Level (all layers), High Profile High-1440 Level (base layer) Video

splice_type	Conditions
0000	splice_decoding_delay = 120 ms; max_splice_rate = 20.0×10^6 bit/s
0001	splice_decoding_delay = 160 ms; max_splice_rate = 15.0×10^6 bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = 10.0×10^6 bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = 9.5×10^6 bit/s
0100-1011	Reserved
1100-1111	User-defined

Table 2-15 – Splice parameters Table 9

High Profile High Level (base layer),
Multi-view Profile Main Level (both layers) Video

splice_type	Conditions
0000	splice_decoding_delay = 120 ms; max_splice_rate = 25.0×10^6 bit/s
0001	splice_decoding_delay = 165 ms; max_splice_rate = 18.0×10^6 bit/s
0010	splice_decoding_delay = 250 ms; max_splice_rate = 12.0×10^6 bit/s
0011-1011	Reserved
1100-1111	User-defined

Table 2-16 – Splice parameters Table 10

High Profile High Level (all layers),
Multi-view Profile High-1440 Level (both layers) Video

splice_type	Conditions
0000	splice_decoding_delay = 120 ms; max_splice_rate = 100.0×10^6 bit/s
0001	splice_decoding_delay = 160 ms; max_splice_rate = 75.0×10^6 bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = 50.0×10^6 bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = 48.0×10^6 bit/s
0100-1011	Reserved
1100-1111	User-defined

Table 2-16-1 – Splice parameters Table 11

4:2:2 Profile Main Level Video

splice_type	Conditions
0000	splice_decoding_delay = 45 ms; max_splice_rate = 50.0×10^6 bit/s
0001	splice_decoding_delay = 90 ms; max_splice_rate = 50.0×10^6 bit/s
0010	splice_decoding_delay = 180 ms; max_splice_rate = 50.0×10^6 bit/s
0011	splice_decoding_delay = 225 ms; max_splice_rate = 40.0×10^6 bit/s
0100	splice_decoding_delay = 250 ms; max_splice_rate = 36.0×10^6 bit/s
0101-1011	Reserved
1100-1111	User-defined

Table 2-16-2 – Splice parameters Table 12

Multi-view Profile Low Level (both layers) Video

splice_type	Conditions
0000	splice_decoding_delay = 115 ms; max_splice_rate = 8.0×10^6 bit/s
0001	splice_decoding_delay = 155 ms; max_splice_rate = 6.0×10^6 bit/s
0010	splice_decoding_delay = 230 ms; max_splice_rate = 4.0×10^6 bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = 3.7×10^6 bit/s
0100-1011	Reserved
1100-1111	User-defined

Table 2-16-3 – Splice parameters Table 13

Multi-view Profile High Level (both layers) Video

splice_type	Conditions
0000	splice_decoding_delay = 120 ms; max_splice_rate = 130.0×10^6 bit/s
0001	splice_decoding_delay = 150 ms; max_splice_rate = 104.0×10^6 bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = 65.0×10^6 bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = 62.4×10^6 bit/s
0100-1011	Reserved
1100-1111	User-defined

Table 2-16-4 _ Splice parameters Table 14

4:2:2 Profile High Level Video

splice_type	Conditions
0000	splice_decoding_delay = 45 ms; max_splice_rate = 300.0×10^6 bit/s
0001	splice_decoding_delay = 90 ms; max_splice_rate = 300.0×10^6 bit/s
0010-0011	Reserved
0100	splice_decoding_delay = 250 ms; max_splice_rate = 180.0×10^6 bit/s
0101-1011	Reserved
1100-1111	User-defined

DTS_next_AU (decoding time stamp next access unit) – This is a 33-bit field, coded in three parts. In the case of continuous and periodic decoding through this splicing point it indicates the decoding time of the first access unit following the splicing point. This decoding time is expressed in the time base which is valid in the Transport Stream packet in which the splice_countdown reaches zero. From the first occurrence of this field onwards, it shall have the same value in all the subsequent Transport Stream packets of the same PID in which it is present, until the packet in which the splice_countdown reaches zero (including this packet).

stuffing_byte – This is a fixed 8-bit value equal to '1111 1111' that can be inserted by the encoder. It is discarded by the decoder.

2.4.3.6 PES packet

See Table 2-17.

2.4.3.7 Semantic definition of fields in PES packet

packet_start_code_prefix – The packet_start_code_prefix is a 24-bit code. Together with the stream_id that follows it constitutes a packet start code that identifies the beginning of a packet. The packet_start_code_prefix is the bit string '0000 0000 0000 0000 0000 0001' (0x000001).

stream_id – In Program Streams, the stream_id specifies the type and number of the elementary stream as defined by the stream_id Table 2-18. In Transport Streams, the stream_id may be set to any valid value which correctly describes the elementary stream type as defined in Table 2-18. In Transport Streams, the elementary stream type is specified in the Program Specific Information as specified in 2.4.4.

PES_packet_length – A 16-bit field specifying the number of bytes in the PES packet following the last byte of the field. A value of 0 indicates that the PES packet length is neither specified nor bounded and is allowed only in PES packets whose payload consists of bytes from a video elementary stream contained in Transport Stream packets.

PES_scrambling_control – The 2-bit PES_scrambling_control field indicates the scrambling mode of the PES packet payload. When scrambling is performed at the PES level, the PES packet header, including the optional fields when present, shall not be scrambled (see Table 2-19).

Table 2-17 – PES packet

Syntax	No. of bits	Mnemonic
PES_packet() {		
packet_start_code_prefix	**24**	**bslbf**
stream_id	**8**	**uimsbf**
PES_packet_length	**16**	**uimsbf**
if (stream_id != program_stream_map		
&& stream_id != padding_stream		
&& stream_id != private_stream_2		
&& stream_id != ECM		
&& stream_id != EMM		
&& stream_id != program_stream_directory		
&& stream_id != DSMCC_stream		
&& stream_id != ITU-T Rec. H.222.1 type E stream) {		
'10'	**2**	**bslbf**
PES_scrambling_control	**2**	**bslbf**
PES_priority	**1**	**bslbf**
data_alignment_indicator	**1**	**bslbf**
copyright	**1**	**bslbf**
original_or_copy	**1**	**bslbf**
PTS_DTS_flags	**2**	**bslbf**
ESCR_flag	**1**	**bslbf**
ES_rate_flag	**1**	**bslbf**
DSM_trick_mode_flag	**1**	**bslbf**
additional_copy_info_flag	**1**	**bslbf**
PES_CRC_flag	**1**	**bslbf**
PES_extension_flag	**1**	**bslbf**
PES_header_data_length	**8**	**uimsbf**
if (PTS_DTS_flags == '10') {		
'0010'	**4**	**bslbf**
PTS [32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		

Table 2-17 – PES packet *(continued)*

Syntax	No. of bits	Mnemonic
if (PTS_DTS_flags == '11') {		
'0011'	4	bslbf
PTS [32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
'0001'	4	bslbf
DTS [32..30]	3	bslbf
marker_bit	1	bslbf
DTS [29..15]	15	bslbf
marker_bit	1	bslbf
DTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		
if (ESCR_flag == '1') {		
reserved	2	bslbf
ESCR_base[32..30]	3	bslbf
marker_bit	1	bslbf
ESCR_base[29..15]	15	bslbf
marker_bit	1	bslbf
ESCR_base[14..0]	15	bslbf
marker_bit	1	bslbf
ESCR_extension	9	uimsbf
marker_bit	1	bslbf
}		
if (ES_rate_flag == '1') {		
marker_bit	1	bslbf
ES_rate	22	uimsbf
marker_bit	1	bslbf
}		
if (DSM_trick_mode_flag == '1') {		
trick_mode_control	3	uimsbf
if (trick_mode_control == fast_forward) {		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
}		
else if (trick_mode_control == slow_motion) {		
rep_cntrl	5	uimsbf
}		
else if (trick_mode_control == freeze_frame) {		
field_id	2	uimsbf
reserved	3	bslbf
}		
else if (trick_mode_control == fast_reverse) {		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
else if (trick_mode_control == slow_reverse) {		
rep_cntrl	5	uimsbf
}		
else		
reserved	5	bslbf
}		
if (additional_copy_info_flag == '1') {		
marker_bit	1	bslbf
additional_copy_info	7	bslbf
}		
if (PES_CRC_flag == '1') {		
previous_PES_packet_CRC	16	bslbf
}		

Table 2-17 – PES packet *(concluded)*

Syntax	No. of bits	Mnemonic
if (PES_extension_flag == '1') {		
PES_private_data_flag	**1**	**bslbf**
pack_header_field_flag	**1**	**bslbf**
program_packet_sequence_counter_flag	**1**	**bslbf**
P-STD_buffer_flag	**1**	**bslbf**
reserved	**3**	**bslbf**
PES_extension_flag_2	**1**	**bslbf**
if (PES_private_data_flag == '1') {		
PES_private_data	**128**	**bslbf**
}		
if (pack_header_field_flag == '1') {		
pack_field_length	**8**	**uimsbf**
pack_header()		
}		
if (program_packet_sequence_counter_flag == '1') {		
marker_bit	**1**	**bslbf**
program_packet_sequence_counter	**7**	**uimsbf**
marker_bit	**1**	**bslbf**
MPEG1_MPEG2_identifier	**1**	**bslbf**
original_stuff_length	**6**	**uimsbf**
}		
if (P-STD_buffer_flag == '1') {		
'01'	**2**	**bslbf**
P-STD_buffer_scale	**1**	**bslbf**
P-STD_buffer_size	**13**	**uimsbf**
}		
if (PES_extension_flag_2 == '1') {		
marker_bit	**1**	**bslbf**
PES_extension_field_length	**7**	**uimsbf**
for (i = 0; i < PES_extension_field_length; i++) {		
reserved	**8**	**bslbf**
}		
}		
}		
for (i = 0; i < N1; i++) {		
stuffing_byte	**8**	**bslbf**
}		
for (i = 0; i < N2; i++) {		
PES_packet_data_byte	**8**	**bslbf**
}		
}		
else if (stream_id == program_stream_map		
|| stream_id == private_stream_2		
|| stream_id == ECM		
|| stream_id == EMM		
|| stream_id == program_stream_directory		
|| stream_id == DSMCC_stream		
|| stream_id == ITU-T Rec. H.222.1 type E stream) {		
for (i = 0; i < PES_packet_length; i++) {		
PES_packet_data_byte	**8**	**bslbf**
}		
}		
else if (stream_id == padding_stream) {		
for (i = 0; i < PES_packet_length; i++) {		
padding_byte	**8**	**bslbf**
}		
}		
}		

Table 2-18 – Stream_id assignments

stream_id	Note	stream coding
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		ISO/IEC 13818-3 or ISO/IEC 11172-3 or ISO/IEC 13818-7 or ISO/IEC 14496-3 audio stream number x xxxx
1110 xxxx		ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 11172-2 or ISO/IEC 14496-2 video stream number xxxx
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Annex A or ISO/IEC 13818-6_DSMCC_stream
1111 0011	2	ISO/IEC_13522_stream
1111 0100	6	ITU-T Rec. H.222.1 type A
1111 0101	6	ITU-T Rec. H.222.1 type B
1111 0110	6	ITU-T Rec. H.222.1 type C
1111 0111	6	ITU-T Rec. H.222.1 type D
1111 1000	6	ITU-T Rec. H.222.1 type E
1111 1001	7	ancillary_stream
1111 1010		ISO/IEC14496-1_SL-packetized_stream
1111 1011		ISO/IEC14496-1_FlexMux_stream
1111 1100 … 1111 1110		reserved data stream
1111 1111	4	program_stream_directory

The notation x means that the values '0' or '1' are both permitted and results in the same stream type. The stream number is given by the values taken by the x's.

NOTE 1 – PES packets of type program_stream_map have unique syntax specified in 2.5.4.1.

NOTE 2 – PES packets of type private_stream_1 and ISO/IEC_13552_stream follow the same PES packet syntax as those for ITU-T Rec. H.262 | ISO/IEC 13818-2 video and ISO/IEC 13818-3 audio streams.

NOTE 3 – PES packets of type private_stream_2, ECM_stream and EMM_stream are similar to private_stream_1 except no syntax is specified after PES_packet_length field.

NOTE 4 – PES packets of type program_stream_directory have a unique syntax specified in 2.5.5.

NOTE 5 – PES packets of type DSM-CC_stream have a unique syntax specified in ISO/IEC 13818- 6.

NOTE 6 – This stream_id is associated with stream_type 0x09 in Table 2-29.

NOTE 7 – This stream_id is only used in PES packets, which carry data from a Program Stream or an ISO/IEC 11172-1 System Stream, in a Transport Stream (refer to 2.4.3.7).

Table 2-19 – PES scrambling control values

Value	Description
00	Not scrambled
01	User-defined
10	User-defined
11	User-defined

PES_priority – This is a 1-bit field indicating the priority of the payload in this PES packet. A '1' indicates a higher priority of the payload of the PES packet payload than a PES packet payload with this field set to '0'. A multiplexor can use the PES_priority bit to prioritize its data within an elementary stream. This field shall not be changed by the transport mechanism.

data_alignment_indicator – This is a 1-bit flag. When set to a value of '1' it indicates that the PES packet header is immediately followed by the video start code or audio syncword indicated in the data_stream_alignment_descriptor in 2.6.10 if this descriptor is present. If set to a value of '1' and the descriptor is not present, alignment as indicated in alignment_type '01' in Table 2-47 and Table 2-48 is required. When set to a value of '0' it is not defined whether any such alignment occurs or not.

copyright – This is a 1-bit field. When set to '1' it indicates that the material of the associated PES packet payload is protected by copyright. When set to '0' it is not defined whether the material is protected by copyright. A copyright descriptor described in 2.6.24 is associated with the elementary stream which contains this PES packet and the copyright flag is set to '1' if the descriptor applies to the material contained in this PES packet.

original_or_copy – This is a 1-bit field. When set to '1' the contents of the associated PES packet payload is an original. When set to '0' it indicates that the contents of the associated PES packet payload is a copy.

PTS_DTS_flags – This is a 2-bit field. When the PTS_DTS_flags field is set to '10', the PTS fields shall be present in the PES packet header. When the PTS_DTS_flags field is set to '11', both the PTS fields and DTS fields shall be present in the PES packet header. When the PTS_DTS_flags field is set to '00' no PTS or DTS fields shall be present in the PES packet header. The value '01' is forbidden.

ESCR_flag – A 1-bit flag, which when set to '1' indicates that ESCR base and extension fields are present in the PES packet header. When set to '0' it indicates that no ESCR fields are present.

ES_rate_flag – A 1-bit flag, which when set to '1' indicates that the ES_rate field is present in the PES packet header. When set to '0' it indicates that no ES_rate field is present.

DSM_trick_mode_flag – A 1-bit flag, which when set to '1' it indicates the presence of an 8-bit trick mode field. When set to '0' it indicates that this field is not present.

additional_copy_info_flag – A 1-bit flag, which when set to '1' indicates the presence of the additional_copy_info field. When set to '0' it indicates that this field is not present.

PES_CRC_flag – A 1-bit flag, which when set to '1' indicates that a CRC field is present in the PES packet. When set to '0' it indicates that this field is not present.

PES_extension_flag – A 1-bit flag, which when set to '1' indicates that an extension field exists in this PES packet header. When set to '0' it indicates that this field is not present.

PES_header_data_length – An 8-bit field specifying the total number of bytes occupied by the optional fields and any stuffing bytes contained in this PES packet header. The presence of optional fields is indicated in the byte that precedes the PES_header_data_length field.

marker_bit – A marker_bit is a 1-bit field that has the value '1'.

PTS (presentation time stamp) – Presentation times shall be related to decoding times as follows: The PTS is a 33-bit number coded in three separate fields. It indicates the time of presentation, $tp_n(k)$, in the system target decoder of a presentation unit k of elementary stream n. The value of PTS is specified in units of the period of the system clock frequency divided by 300 (yielding 90 kHz). The presentation time is derived from the PTS according to equation 2-11 below. Refer to 2.7.4 for constraints on the frequency of coding presentation timestamps.

$$PTS(k) = ((system\_clock\_frequency \times tp_n(k)) \; DIV \; 300) \; \% \; 2^{33} \qquad (2\text{-}11)$$

where $tp_n(k)$ is the presentation time of presentation unit $P_n(k)$.

In the case of audio, if a PTS is present in PES packet header it shall refer to the first access unit commencing in the PES packet. An audio access unit commences in a PES packet if the first byte of the audio access unit is present in the PES packet.

In the case of video, if a PTS is present in a PES packet header it shall refer to the access unit containing the first picture start code that commences in this PES packet. A picture start code commences in PES packet if the first byte of the picture start code is present in the PES packet.

For audio presentation units (PUs), video PUs in low_delay sequences, and B-pictures, the presentation time $tp_n(k)$ shall be equal to the decoding time $td_n(k)$.

For I- and P-pictures in non-low_delay sequences and in the case when there is no decoding discontinuity between access units (AUs) k and k', the presentation time $tp_n(k)$ shall be equal to the decoding time $td_n(k')$ of the next transmitted I- or P-picture (refer to 2.7.5). If there is a decoding discontinuity, or the stream ends, the difference between $tp_n(k)$ and $td_n(k)$ shall be the same as if the original stream had continued without a discontinuity and without ending.

NOTE 1 – A low_delay sequence is a video sequence in which the low_delay flag is set (refer to 6.2.2.3 of ITU-T Rec. H.262 | ISO/IEC 13818-2).

If there is filtering in audio, it is assumed by the system model that filtering introduces no delay, hence the sample referred to by PTS at encoding is the same sample referred to by PTS at decoding. In the case of scalable coding refer to 2.7.6.

DTS (decoding time stamp) – The DTS is a 33-bit number coded in three separate fields. It indicates the decoding time, $td_n(j)$, in the system target decoder of an access unit j of elementary stream n. The value of DTS is specified in units of the period of the system clock frequency divided by 300 (yielding 90 kHz). The decoding time derived from the DTS according to equation 2-12 below:

$$DTS(j) = ((system\_clock\_frequency \times td_n(j)) \; DIV \; 300) \; \% \; 2^{33} \tag{2-12}$$

where $td_n(j)$ is the decoding time of access unit $A_n(j)$.

In the case of video, if a DTS is present in a PES packet header it shall refer to the access unit containing the first picture start code that commences in this PES packet. A picture start code commences in PES packet if the first byte of the picture start code is present in the PES packet.

In the case of scalable coding refer to 2.7.6.

ESCR_base; ESCR_extension – The elementary stream clock reference is a 42-bit field coded in two parts. The first part, ESCR_base, is a 33-bit field whose value is given by ESCR_base(i), as given in equation 2-14. The second part, ESCR_ext, is a 9-bit field whose value is given by ESCR_ext(i), as given in equation 2-15. The ESCR field indicates the intended time of arrival of the byte containing the last bit of the ESCR_base at the input of the PES-STD for PES streams (refer to 2.5.2.4).

Specifically:

$$ESCR(i) = ESCR\_base(i) \times 300 + ESCR\_ext(i) \tag{2-13}$$

where

$$ESCR\_base(i) = ((system\_clock\_frequency \times t(i)) \; DIV \; 300) \; \% \; 2^{33} \tag{2-14}$$

$$ESCR\_ext(i) = ((system\_clock\_frequency \times t(i)) \; DIV \; 1) \; \% \; 300 \tag{2-15}$$

The ESCR and ES_rate field (refer to semantics immediately following) contain timing information relating to the sequence of PES streams. These fields shall satisfy the constraints defined in 2.7.3.

ES_rate (elementary stream rate) – The ES_rate field is a 22-bit unsigned integer specifying the rate at which the system target decoder receives bytes of the PES packet in the case of a PES stream. The ES_rate is valid in the PES packet in which it is included and in subsequent PES packets of the same PES stream until a new ES_rate field is encountered. The value of the ES_rate is measured in units of 50 bytes/second. The value 0 is forbidden. The value of the ES_rate is used to define the time of arrival of bytes at the input of a P-STD for PES streams defined in 2.5.2.4. The value encoded in the ES_rate field may vary from PES_packet to PES_packet.

trick_mode_control – A 3-bit field that indicates which trick mode is applied to the associated video stream. In cases of other types of elementary streams, the meanings of this field and those defined by the following five bits are undefined. For the definition of trick_mode status, refer to the **trick mode** section of 2.4.2.3.

When trick_mode status is false, the number of times N, a picture is output by the decoding process for progressive sequences, is specified for each picture by the repeat_first_field and top_field_first fields in the case of ITU-T Rec. H.262 | ISO/IEC 13818-2 Video, and is specified through the sequence header in the case of ISO/IEC 11172-2 Video.

For interlaced sequences, when trick_mode status is false, the number of times N, a picture is output by the decoding process for progressive sequences, is specified for each picture by the repeat_first_field and progressive_frame fields in the case of ITU-T Rec. H.262 | ISO/IEC 13818-2 Video.

When trick mode status is true, the number of times that a picture shall be displayed depends on the value of N.

When the value of this field changes or trick mode operations cease, any combination of the following may occur:

- discontinuity in the time base;
- decoding discontinuity;
- continuity counter discontinuity.

Table 2-20 – Trick mode control values

Value	Description
'000'	Fast forward
'001'	Slow motion
'010'	Freeze frame
'011'	Fast reverse
'100'	Slow reverse
'101'-'111'	Reserved

In the context of trick mode, the non-normal speed of decoding and presentation may cause the values of certain fields defined in video elementary stream data to be incorrect. Likewise, the semantic constraint on the slice structure may be invalid. The video syntax elements to which this exception applies are:

- bit_rate;
- vbv_delay;
- repeat_first_field;
- v_axis_positive;
- field_sequence;
- subcarrier;
- burst_amplitude;
- subcarrier_phase.

A decoder cannot rely on the values encoded in these fields when in trick mode.

Decoders are not normatively required to decode the trick_mode_control field. However, the following normative requirements shall apply to decoders that do decode the trick_mode_control field.

fast forward – The value '000', in the trick_mode_control field. When this value is present it indicates a fast forward video stream and defines the meaning of the following five bits in the PES packet header. The intra_slice_refresh bit may be set to '1' indicating that there may be missing macroblocks which the decoder may replace with co-sited macroblocks of previously decoded pictures. The field_id field, defined in Table 2-21, indicates which field or fields should be displayed. The frequency_truncation field indicates that a restricted set of coefficients may be included. The meaning of the values of this field are shown in Table 2-22.

slow motion – The value '001', in the trick_mode_control field. When this value is present it indicates a slow motion video stream and defines the meaning of the following five bits in the PES packet header. In the case of progressive sequences, the picture should be displayed N × rep_cntrl times, where N is defined above.

In the case of ISO/IEC 11172-2 Video and ITU-T Rec. H.262 | ISO/IEC 13818-2 Video progressive sequences, the picture should be displayed for N × rep_cntrl picture duration.

In the case of ITU-T Rec. H.262 | ISO/IEC 13818-2 interlaced sequences, the picture should be displayed for N × rep_cntrl field duration. If the picture is a frame picture, the first field to be displayed is the top field if top_field_first is 1, and the bottom field if top_field_first is '0' (refer to ITU-T Rec. H.262 | ISO/IEC 13818-2). This field is displayed for N × rep_cntrl / 2 field duration. The other field of the picture is then displayed for N – N × rep_cntrl / 2 field duration.

freeze frame – The value '010', in the trick_mode_control field. When this value is present it indicates a freeze frame video stream and defines the meaning of the following five bits in the PES packet header. The field_id field, defined in Table 2-21, identifies which field(s) should be displayed. The field_id field refers to the first video access unit that commences in the PES packet which contains the field_id field, unless the PES packet contains zero payload bytes. In the latter case the field_id field refers to the most recent previous video access unit.

fast reverse – The value '011', in the trick_mode_control field. When this value is present it indicates a fast reverse video stream and defines the meaning of the following five bits in the PES packet header. The intra_slice_refresh bit may be set to '1' indicating that there may be missing macroblocks which the decoder may replace with co-sited macroblocks of previously decoded pictures. The field_id field, defined in Table 2-21, indicates which field or fields should be displayed. The frequency_truncation field indicates that a restricted set of coefficients may be included. The meaning of the values of this field are shown in Table 2-22, "Coefficient selection values".

slow reverse – The value '100', in the trick_mode_control field. When this value is present it indicates a slow reverse video stream and defines the meaning of the following five bits in the PES packet header. In the case of ISO/IEC 11172-2 Video and ITU-T Rec. H.262 | ISO/IEC 13818-2 Video progressive sequences, the picture should be displayed for N × rep_cntrl picture duration, where N is defined above.

In the case of ITU-T Rec. H.262 | ISO/IEC 13818-2 interlaced sequences, the picture should be displayed for N × rep_cntrl field duration. If the picture is a frame picture, the first field to be displayed is the bottom field if top_field_first is 1, and the top field if top_field_first is '0' (refer to ITU-T Rec. H.262 | ISO/IEC 13818-2). This field is displayed for N × rep_cntrl / 2 field duration. The other field of the picture is then displayed for N – N × rep_cntrl / 2 field duration.

field_id – A 2-bit field that indicates which field(s) should be displayed. It is coded according to Table 2-21.

Table 2-21 – Field_id field control values

Value	Description
'00'	Display from top field only
'01'	Display from bottom field only
'10'	Display complete frame
'11'	Reserved

intra_slice_refresh – A 1-bit flag, which when set to '1', indicates that there may be missing macroblocks between coded slices of video data in this PES packet. When set to '0' this may not occur. For more information see ITU-T Rec. H.262 | ISO/IEC 13818-2. The decoder may replace missing macroblocks with co-sited macroblocks of previously decoded pictures.

frequency_truncation – A 2-bit field which indicates that a restricted set of coefficients may have been used in coding the video data in this PES packet. The values are defined in Table 2-22.

Table 2-22 – Coefficient selection values

Value	Description
'00'	Only DC coefficients are non-zero
'01'	Only the first three coefficients are non-zero
'10'	Only the first six coefficients are non-zero
'11'	All coefficients may be non-zero

rep_cntrl – A 5-bit field that indicates the number of times each field in an interlaced picture should be displayed, or the number of times that a progressive picture should be displayed. It is a function of the trick_mode_control field and the top_field_first bit in the video sequence header whether the top field or the bottom field should be displayed first in the case of interlaced pictures. The value '0' is forbidden.

additional_copy_info – This 7-bit field contains private data relating to copyright information.

previous_PES_packet_CRC – The previous_PES_packet_CRC is a 16-bit field that contains the CRC value that yields a zero output of the 16 registers in the decoder similar to the one defined in Annex A, but with the polynomial:

$$x^{16} + x^{12} + x^5 + 1$$

after processing the data bytes of the previous PES packet, exclusive of the PES packet header.

NOTE 2 – This CRC is intended for use in network maintenance such as isolating the source of intermittent errors. It is not intended for use by elementary stream decoders. It is calculated only over the data bytes because PES packet header data can be modified during transport.

PES_private_data_flag – A 1-bit flag which when set to '1' indicates that the PES packet header contains private data. When set to a value of '0' it indicates that private data is not present in the PES header.

pack_header_field_flag – A 1-bit flag which when set to '1' indicates that an ISO/IEC 11172-1 pack header or a Program Stream pack header is stored in this PES packet header. If this field is in a PES packet that is contained in a Program Stream, then this field shall be set to '0'. In a Transport Stream, when set to the value '0' it indicates that no pack header is present in the PES header.

program_packet_sequence_counter_flag – A 1-bit flag which when set to '1' indicates that the program_packet_sequence_counter, MPEG1_MPEG2_identifier, and original_stuff_length fields are present in this PES packet. When set to a value of '0' it indicates that these fields are not present in the PES header.

P-STD_buffer_flag – A 1-bit flag which when set to '1' indicates that the P-STD_buffer_scale and P-STD_buffer_size are present in the PES packet header. When set to a value of '0' it indicates that these fields are not present in the PES header.

PES_extension_flag_2 – A 1-bit field which when set to '1' indicates the presence of the PES_extension_field_length field and associated fields. When set to a value of '0' this indicates that the PES_extension_field_length field and any associated fields are not present.

PES_private_data – This is a 16-byte field which contains private data. This data, combined with the fields before and after, shall not emulate the packet_start_code_prefix (0x000001).

pack_field_length – This is an 8-bit field which indicates the length, in bytes, of the pack_header_field().

program_packet_sequence_counter – The program_packet_sequence_counter field is a 7-bit field. It is an optional counter that increments with each successive PES packet from a Program Stream or from an ISO/IEC 11172-1 Stream or the PES packets associated with a single program definition in a Transport Stream, providing functionality similar to a continuity counter (refer to 2.4.3.2). This allows an application to retrieve the original PES packet sequence of a Program Stream or the original packet sequence of the original ISO/IEC 11172-1 stream. The counter will wrap around to 0 after its maximum value. Repetition of PES packets shall not occur. Consequently, no two consecutive PES packets in the program multiplex shall have identical program_packet_sequence_counter values.

MPEG1_MPEG2_identifier – A 1-bit flag which when set to '1' indicates that this PES packet carries information from an ISO/IEC 11172-1 stream. When set to '0' it indicates that this PES packet carries information from a Program Stream.

original_stuff_length – This 6-bit field specifies the number of stuffing bytes used in the original ITU-T Rec. H.222.0 | ISO/IEC 13818-1 PES packet header or in the original ISO/IEC 11172-1 packet header.

P-STD_buffer_scale – The P-STD_buffer_scale is a 1-bit field, the meaning of which is only defined if this PES packet is contained in a Program Stream. It indicates the scaling factor used to interpret the subsequent P-STD_buffer_size field. If the preceding stream_id indicates an audio stream, P-STD_buffer_scale shall have the value '0'. If the preceding stream_id indicates a video stream, P-STD_buffer_scale shall have the value '1'. For all other stream types, the value may be either '1' or '0'.

P-STD_buffer_size – The P-STD_buffer_size is a 13-bit unsigned integer, the meaning of which is only defined if this PES packet is contained in a Program Stream. It defines the size of the input buffer, BS_n, in the P-STD. If P-STD_buffer_scale has the value '0', then the P-STD_buffer_size measures the buffer size in units of 128 bytes. If P-STD_buffer_scale has the value '1', then the P-STD_buffer_size measures the buffer size in units of 1024 bytes. Thus:

$$\text{if } (P\text{-}STD\_buffer\_scale == 0)$$
$$BS_n = \bar{P} - STD\_buffer\_size \times 128 \tag{2-16}$$

else

$$BS_n = P - STD\_buffer\_size \times 1024 \tag{2-17}$$

The encoded value of the P-STD buffer size takes effect immediately when the P-STD_buffer_size field is received by the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 System Target Decoder (refer to 2.7.7).

PES_extension_field_length – This is a 7-bit field which specifies the length, in bytes, of the data following this field in the PES extension field up to and including any reserved bytes.

stuffing_byte – This is a fixed 8-bit value equal to '1111 1111' that can be inserted by the encoder, for example to meet the requirements of the channel. It is discarded by the decoder. No more than 32 stuffing bytes shall be present in one PES packet header.

PES_packet_data_byte – PES_packet_data_bytes shall be contiguous bytes of data from the elementary stream indicated by the packet's stream_id or PID. When the elementary stream data conforms to ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 13818-3, the PES_packet_data_bytes shall be byte aligned to the bytes of this Recommendation | International Standard. The byte-order of the elementary stream shall be preserved. The number of PES_packet_data_bytes, N, is specified by the PES_packet_length field. N shall be equal to the value indicated in the PES_packet_length minus the number of bytes between the last byte of the PES_packet_length field and the first PES_packet_data_byte.

In the case of a private_stream_1, private_stream_2, ECM_stream, or EMM_stream, the contents of the PES_packet_data_byte field are user definable and will not be specified by ITU-T | ISO/IEC in the future.

padding_byte – This is a fixed 8-bit value equal to '1111 1111'. It is discarded by the decoder.

2.4.3.8 Carriage of Program Streams and ISO/IEC 11172-1 Systems streams in the Transport Stream

The Transport Stream contains optional fields to support the carriage of Program Streams and ISO/IEC 11172-1 Systems streams, in a way that allows simple reconstruction of the respective stream at the decoder.

When placing a Program Stream into a Transport Stream, Program Stream PES packets with stream_id values of private_stream_1, ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 11172-2 video, and ISO/IEC 13818-3 or ISO/IEC 11172-3 audio, are carried in Transport Stream packets.

For these PES packets, when reconstructing the Program Stream at the Transport Stream decoder, the PES packet data is copied to the Program Stream being reconstructed.

For Program Streams PES packets with stream_id values of program_stream_map, padding_stream, private_stream_2, ECM, EMM, DSM_CC_stream, or program_stream_directory, all the bytes of the Program Stream PES packet, except for the packet_start_code_prefix, are placed into the data_bytes fields of a new PES packet. The stream_id of this new PES packet has the value of ancillary_stream (refer to Table 2-18). This new PES packet is then carried in Transport Stream packets.

When reconstructing the Program Stream at the Transport Stream decoder, for PES packets with a stream_id value of ancillary_stream_id, packet_start_code_prefix is written to the Program Stream being reconstructed, followed by the data_byte fields from these Transport Stream PES packets.

ISO/IEC 11172-1 streams are carried within Transport Streams by first replacing ISO/IEC 11172-1 packet headers with ITU-T Rec. H.262 | ISO/IEC 13818-2 PES packet headers. ISO/IEC 11172-1 packet header field values are copied to the equivalent ITU-T Rec. H.262 | ISO/IEC 13818-2 PES packet header fields.

The program_packet_sequence_counter field is included within the header of each PES packet carrying data from a Program Stream, or an ISO/IEC 11172-1 System stream. This allows the order of PES packets in the original Program Stream, or packets in the original ISO/IEC 11172-1 System stream, to be reproduced at the decoder.

The pack_header() field of a Program Stream, or an ISO/IEC 11172-1 System stream, is carried in the Transport Stream in the header of the immediately following PES packet.

2.4.4 Program specific information

Program Specific Information (PSI) includes both ITU-T Rec. H.222.0 | ISO/IEC 13818-1 normative data and private data that enable demultiplexing of programs by decoders. Programs are composed of one or more elementary streams, each labelled with a PID. Programs, elementary streams or parts thereof may be scrambled for conditional access. However, Program Specific Information shall not be scrambled.

In Transport Streams, Program Specific Information is classified into five table structures as shown in Table 2-23. While these structures may be thought of as simple tables, they shall be segmented into sections and inserted in Transport Stream packets, some with predetermined PIDs and others with user selectable PIDs.

Table 2-23 – Program specific information

Structure Name	Stream Type	PID number	Description
Program Association Table	ITU-T Rec. H.222.0 | ISO/IEC 13818-1	0x00	Associates Program Number and Program Map Table PID
Program Map Table	ITU-T Rec. H.222.0 | ISO/IEC 13818-1	Assignment indicated in the PAT	Specifies PID values for components of one or more programs
Network Information Table	Private	Assignment indicated in the PAT	Physical network parameters such as FDM frequencies, Transponder Numbers, etc.
Conditional Access Table	ITU-T Rec. H.222.0 | ISO/IEC 13818-1	0x01	Associates one or more (private) EMM streams each with a unique PID value
Transport Stream Description Table	ITU-T Rec. H.222.0 | ISO/IEC 13818-1	0x02	Associates one or more descriptors from Table 2-39 to an entire Transport Stream

ITU-T Rec. H.222.0 | ISO/IEC 13818-1 defined PSI tables shall be segmented into one or more sections that are carried within transports packets. A section is a syntactic structure that shall be used for mapping each ITU-T Rec. H.222.0 | ISO/IEC 13818-1 defined PSI table into Transport Stream packets.

Along with ITU-T Rec. H.222.0 | ISO/IEC 13818-1 defined PSI tables, it is possible to carry private data tables. The means by which private information is carried within Transport Stream packets is not defined by this Specification. It may be structured in the same manner used for carrying of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 defined PSI tables, such that the syntax for mapping this private data is identical to that used for the mapping of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 defined PSI tables. For this purpose, a private section is defined. If the private data is carried in Transport Stream packets with the same PID value as Transport Stream packets carrying Program Map Tables, (as identified in the Program Association Table), then the private_section syntax and semantics shall be used. The data carried in the private_data_bytes may be scrambled. However, no other fields of the private_section shall be scrambled. This private_section allows data to be transmitted with a minimum of structure. When this structure is not used, the mapping of private data within Transport Stream packets is not defined by this Recommendation | International Standard.

Sections may be variable in length. The beginning of a section is indicated by a pointer_field in the Transport Stream packet payload. The syntax of this field is specified in Table 2-24.

Adaptation fields may occur in Transport Stream packets carrying PSI sections.

Within a Transport Stream, packet stuffing bytes of value 0xFF may be found in the payload of Transport Stream packets carrying PSI and/or private_sections only after the last byte of a section. In this case all bytes until the end of the Transport Stream packet shall also be stuffing bytes of value 0xFF. These bytes may be discarded by a decoder. In such a case, the payload of the next Transport Stream packet with the same PID value shall begin with a pointer_field of value 0x00 indicating that the next section starts immediately thereafter.

Each Transport Stream shall contain one or more Transport Stream packets with PID value 0x0000. These Transport Stream packets together shall contain a complete Program Association Table, providing a complete list of all programs within the Transport Stream. The most recently transmitted version of the table with the current_next_indicator set to a value of '1' shall always apply to the current data in the Transport Stream. Any changes in the programs carried within the Transport Stream shall be described in an updated version of the Program Association Table carried in Transport Stream packets with PID value 0x0000. These sections shall all use table_id value 0x00. Only sections with this value of table_id are permitted within Transport Stream packets with PID value of 0x0000. For a new version of the PAT to become valid, all sections (as indicated in the last_section_number) with a new version_number and with the current_next_indicator set to '1' must exit B_{sys} defined in the T-STD (refer to 2.4.2). The PAT becomes valid when the last byte of the section needed to complete the table exits B_{sys}.

Whenever one or more elementary streams within a Transport Stream are scrambled, Transport Stream packets with a PID value 0x0001 shall be transmitted containing a complete Conditional Access Table including CA_descriptors associated with the scrambled streams. The transmitted Transport Stream packets will together form one complete version of the conditional access table. The most recently transmitted version of the table with the current_next_indicator set to a value of '1' shall always apply to the current data in the Transport Stream. Any changes in scrambling making the existing table invalid or incomplete shall be described in an updated version of the conditional access table. These sections will all use table_id value 0x01. Only sections with this table_id value are permitted within Transport Stream packets with a PID value of 0x0001. For a new version of the CAT to become valid, all sections (as indicated in the last_section_number) with a new version_number and with the current_next_indicator set to '1' must exit B_{sys}. The CAT becomes valid when the last byte of the section needed to complete the table exits B_{sys}.

Each Transport Stream shall contain one or more Transport Stream packets with PID values which are labelled under the program association table as Transport Stream packets containing TS program map sections. Each program listed in the Program Association Table shall be described in a unique TS_program_map_section. Every program shall be fully defined within the Transport Stream itself. Private data which has an associated elementary_PID field in the appropriate Program Map Table section is part of the program. Other private data may exist in the Transport Stream without being listed in the Program Map Table section. The most recently transmitted version of the TS_program_map_section with the current_next_indicator set to a value of '1' shall always apply to the current data within the Transport Stream. Any changes in the definition of any of the programs carried within the Transport Stream shall be described in an updated version of the corresponding section of the program map table carried in Transport Stream packets with the PID value identified as the program_map_PID for that specific program. All Transport Stream packets which carry a given TS_program_map_section shall have the same PID value. During the continuous existence of a program, including all of its associated events, the program_map_PID shall not change. A program definition shall not span more than one TS_program_map_section. A new version of a TS_program_map_section becomes valid when the last byte of that section with a new version_number and with the current_next_indicator set to '1' exits B_{sys}.

Sections with a table_id value of 0x02 shall contain Program Map Table information. Such sections may be carried in Transport Stream packets with different PID values.

The Network Information Table is optional and its contents are private. If present it is carried within Transport Stream packets that will have the same PID value, called the network_PID. The network_PID value is defined by the user and, when present, shall be found in the Program Association Table under the reserved program_number 0x0000. If the network information table exists, it shall take the form of one or more private_sections.

The maximum number of bytes in a section of a ITU-T Rec. H.222.0 | ISO/IEC 13818-1 defined PSI table is 1024 bytes. The maximum number of bytes in a private_section is 4096 bytes.

The Transport Stream Description Table is optional. When present, the Transport Stream Description is carried within Transport Stream packets that have a PID value 0x0002 as specified in Table 2-23 and shall apply to the entire Transport Stream. Sections of the Transport Stream Description shall use a table_id value of 0x03 as specified in Table 2-26 and its contents are restricted to descriptors specified in Table 2-39. The TS_description_section becomes valid when the last byte of the section required to complete the table exits B_{sys}.

There are no restrictions on the occurrence of start codes, sync bytes or other bit patterns in PSI data, whether this Recommendation | International Standard or private.

2.4.4.1 Pointer

The pointer_field syntax is defined in Table 2-24.

Table 2-24 – Program specific information pointer

Syntax	No. of bits	Mnemonic
pointer_field	**8**	**uimsbf**

2.4.4.2 Semantics definition of fields in pointer syntax

pointer_field – This is an 8-bit field whose value shall be the number of bytes, immediately following the pointer_field until the first byte of the first section that is present in the payload of the Transport Stream packet (so a value of 0x00 in the pointer_field indicates that the section starts immediately after the pointer_field). When at least one section begins in a given Transport Stream packet, then the payload_unit_start_indicator (refer to 2.4.3.2) shall be set to 1 and the first byte of the payload of that Transport Stream packet shall contain the pointer. When no section begins in a given Transport Stream packet, then the payload_unit_start_indicator shall be set to 0 and no pointer shall be sent in the payload of that packet.

2.4.4.3 Program association Table

The Program Association Table provides the correspondence between a program_number and the PID value of the Transport Stream packets which carry the program definition. The program_number is the numeric label associated with a program.

The overall table is contained in one or more sections with the following syntax. It may be segmented to occupy multiple sections (see Table 2.25).

Table 2-25 – Program association section

Syntax	No. of bits	Mnemonic
program_association_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
reserved	**2**	**bslbf**
section_length	**12**	**uimsbf**
transport_stream_id	**16**	**uimsbf**
reserved	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
program_number	**16**	**uimsbf**
reserved	**3**	**bslbf**
if (program_number == '0') {		
network_PID	**13**	**uimsbf**
}		
else {		
program_map_PID	**13**	**uimsbf**
}		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.4 Table_id assignments

The table_id field identifies the contents of a Transport Stream PSI section as shown in Table 2-26.

Table 2-26 – table_id assignment values

Value	description
0x00	program_association_section
0x01	conditional_access_section (CA_section)
0x02	TS_program_map_section
0x03	TS_description_section
0x04	ISO_IEC_14496_scene_description_section
0x05	ISO_IEC_14496_object_descriptor_section
0x06-0x37	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 reserved
0x38-0x3F	Defined in ISO/IEC 13818-6
0x40-0xFE	User private
0xFF	forbidden

2.4.4.5 Semantic definition of fields in program association section

table_id – This is an 8-bit field, which shall be set to 0x00 as shown in Table 2-26.

section_syntax_indicator – The section_syntax_indicator is a 1-bit field which shall be set to '1'.

section_length – This is a 12-bit field, the first two bits of which shall be '00'. The remaining 10 bits specify the number of bytes of the section, starting immediately following the section_length field, and including the CRC. The value in this field shall not exceed 1021 (0x3FD).

transport_stream_id – This is a 16-bit field which serves as a label to identify this Transport Stream from any other multiplex within a network. Its value is defined by the user.

version_number – This 5-bit field is the version number of the whole Program Association Table. The version number shall be incremented by 1 modulo 32 whenever the definition of the Program Association Table changes. When the current_next_indicator is set to '1', then the version_number shall be that of the currently applicable Program Association Table. When the current_next_indicator is set to '0', then the version_number shall be that of the next applicable Program Association Table.

current_next_indicator – A 1-bit indicator, which when set to '1' indicates that the Program Association Table sent is currently applicable. When the bit is set to '0', it indicates that the table sent is not yet applicable and shall be the next table to become valid.

section_number – This 8-bit field gives the number of this section. The section_number of the first section in the Program Association Table shall be 0x00. It shall be incremented by 1 with each additional section in the Program Association Table.

last_section_number – This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the complete Program Association Table.

program_number – Program_number is a 16-bit field. It specifies the program to which the program_map_PID is applicable. When set to 0x0000, then the following PID reference shall be the network PID. For all other cases the value of this field is user defined. This field shall not take any single value more than once within one version of the Program Association Table.

NOTE – The program_number may be used as a designation for a broadcast channel, for example.

network_PID – The network_PID is a 13-bit field, which is used only in conjunction with the value of the program_number set to 0x0000, specifies the PID of the Transport Stream packets which shall contain the Network Information Table. The value of the network_PID field is defined by the user, but shall only take values as specified in Table 2-3. The presence of the network_PID is optional.

program_map_PID – The program_map_PID is a 13-bit field specifying the PID of the Transport Stream packets which shall contain the program_map_section applicable for the program as specified by the program_number. No program_number shall have more than one program_map_PID assignment. The value of the program_map_PID is defined by the user, but shall only take values as specified in Table 2-3.

CRC_32 – This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex A after processing the entire program association section.

2.4.4.6 Conditional access Table

The Conditional Access (CA) Table provides the association between one or more CA systems, their EMM streams and any special parameters associated with them. Refer to 2.6.16 for a definition of the descriptor() field in Table 2-27.

The table is contained in one or more sections with the following syntax. It may be segmented to occupy multiple sections.

Table 2-27 – Conditional access section

Syntax	No. of bits	Mnemonic
CA_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
reserved	**2**	**bslbf**
section_length	**12**	**uimsbf**
reserved	**18**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.7 Semantic definition of fields in conditional access section

table_id – This is an 8-bit field, which shall be set to 0x01 as specified in Table 2-26.

section_syntax_indicator – The section_syntax_indicator is a 1-bit field which shall be set to '1'.

section_length – This is a 12-bit field, the first two bits of which shall be '00'. The remaining 10-bits specify the number of bytes of the section starting immediately following the section_length field, and including the CRC. The value in this field shall not exceed 1021 (0x3FD).

version_number – This 5-bit field is the version number of the entire conditional access table. The version number shall be incremented by 1 modulo 32 when a change in the information carried within the CA table occurs. When the current_next_indicator is set to '1', then the version_number shall be that of the currently applicable Conditional Access Table. When the current_next_indicator is set to '0', then the version_number shall be that of the next applicable Conditional Access Table.

current_next_indicator – A 1-bit indicator, which when set to '1' indicates that the Conditional Access Table sent is currently applicable. When the bit is set to '0', it indicates that the Conditional Access Table sent is not yet applicable and shall be the next Conditional Access Table to become valid.

section_number – This 8-bit field gives the number of this section. The section_number of the first section in the Conditional Access Table shall be 0x00. It shall be incremented by 1 with each additional section in the Conditional Access Table.

last_section_number – This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the Conditional Access Table.

CRC_32 – This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex A after processing the entire conditional access section.

2.4.4.8 Program Map Table

The Program Map Table provides the mappings between program numbers and the program elements that comprise them. A single instance of such a mapping is referred to as a "program definition". The program map table is the complete collection of all program definitions for a Transport Stream. This table shall be transmitted in packets, the PID values of which are selected by the encoder. More than one PID value may be used, if desired. The table is contained in one or more sections with the following syntax. It may be segmented to occupy multiple sections. In each section, the section number field shall be set to zero. Sections are identified by the program_number field.

Definition for the descriptor() fields may be found in 2.6 (see Table 2-28).

Table 2-28 – Transport Stream program map section

Syntax	No. of bits	Mnemonic
TS_program_map_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
reserved	**2**	**bslbf**
section_length	**12**	**uimsbf**
program_number	**16**	**uimsbf**
reserved	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
reserved	**3**	**bslbf**
PCR_PID	**13**	**uimsbf**
reserved	**4**	**bslbf**
program_info_length	**12**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
for (i = 0; i < N1; i++) {		
stream_type	**8**	**uimsbf**
reserved	**3**	**bslbf**
elementary_PID	**13**	**uimsbf**
reserved	**4**	**bslbf**
ES_info_length	**12**	**uimsbf**
for (i = 0; i < N2; i++) {		
descriptor()		
}		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.9 Semantic definition of fields in Transport Stream program map section

table_id – This is an 8-bit field, which in the case of a TS_program_map_section shall be always set to 0x02 as shown in Table 2-26.

section_syntax_indicator – The section_syntax_indicator is a 1-bit field which shall be set to '1'.

section_length – This is a 12-bit field, the first two bits of which shall be '00'. The remaining 10 bits specify the number of bytes of the section starting immediately following the section_length field, and including the CRC. The value in this field shall not exceed 1021 (0x3FD).

program_number – program_number is a 16-bit field. It specifies the program to which the program_map_PID is applicable. One program definition shall be carried within only one TS_program_map_section. This implies that a program definition is never longer than 1016 (0x3F8). See Informative Annex C for ways to deal with the cases when that length is not sufficient. The program_number may be used as a designation for a broadcast channel, for example. By describing the different program elements belonging to a program, data from different sources (e.g. sequential events) can be concatenated together to form a continuous set of streams using a program_number. For examples of applications refer to Annex C.

version_number – This 5-bit field is the version number of the TS_program_map_section. The version number shall be incremented by 1 modulo 32 when a change in the information carried within the section occurs. Version number refers to the definition of a single program, and therefore to a single section. When the current_next_indicator is set to '1', then the version_number shall be that of the currently applicable TS_program_map_section. When the current_next_indicator is set to '0', then the version_number shall be that of the next applicable TS_program_map_section.

current_next_indicator – A 1-bit field, which when set to '1' indicates that the TS_program_map_section sent is currently applicable. When the bit is set to '0', it indicates that the TS_program_map_section sent is not yet applicable and shall be the next TS_program_map_section to become valid.

section_number – The value of this 8-bit field shall be 0x00.

last_section_number – The value of this 8-bit field shall be 0x00.

PCR_PID – This is a 13-bit field indicating the PID of the Transport Stream packets which shall contain the PCR fields valid for the program specified by program_number. If no PCR is associated with a program definition for private streams, then this field shall take the value of 0x1FFF. Refer to the semantic definition of PCR in 2.4.3.5 and Table 2-3 for restrictions on the choice of PCR_PID value.

program_info_length – This is a 12-bit field, the first two bits of which shall be '00'. The remaining 10 bits specify the number of bytes of the descriptors immediately following the program_info_length field.

stream_type – This is an 8-bit field specifying the type of program element carried within the packets with the PID whose value is specified by the elementary_PID. The values of stream_type are specified in Table 2-29.

NOTE – An ITU-T Rec. H.222.0 | ISO/IEC 13818-1 auxiliary stream is available for data types defined by this Specification, other than audio, video, and DSM CC, such as Program Stream Directory and Program Stream Map.

elementary_PID – This is a 13-bit field specifying the PID of the Transport Stream packets which carry the associated program element.

ES_info_length – This is a 12-bit field, the first two bits of which shall be '00'. The remaining 10 bits specify the number of bytes of the descriptors of the associated program element immediately following the ES_info_length field.

CRC_32 – This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex B after processing the entire Transport Stream program map section.

2.4.4.10 Syntax of the Private section

When private data is sent in Transport Stream packets with a PID value designated as a Program Map Table PID in the Program Association Table the private_section shall be used. The private_section allows data to be transmitted with a minimum of structure while enabling a decoder to parse the stream. The sections may be used in two ways: if the section_syntax_indicator is set to '1', then the whole structure common to all tables shall be used; if the indicator is set to '0', then only the fields 'table_id' through 'private_section_length' shall follow the common structure syntax and semantics and the rest of the private_section may take any form the user determines. Examples of extended use of this syntax are found in Informative Annex C.

Table 2-29 – Stream type assignments

Value	Description
0x00	ITU-T | ISO/IEC Reserved
0x01	ISO/IEC 11172 Video
0x02	ITU-T Rec. H.262 | ISO/IEC 13818-2 Video or ISO/IEC 11172-2 constrained parameter video stream
0x03	ISO/IEC 11172 Audio
0x04	ISO/IEC 13818-3 Audio
0x05	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 private_sections
0x06	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 PES packets containing private data
0x07	ISO/IEC 13522 MHEG
0x08	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Annex A DSM-CC
0x09	ITU-T Rec. H.222.1
0x0A	ISO/IEC 13818-6 type A
0x0B	ISO/IEC 13818-6 type B
0x0C	ISO/IEC 13818-6 type C
0x0D	ISO/IEC 13818-6 type D
0x0E	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 auxiliary
0x0F	ISO/IEC 13818-7 Audio with ADTS transport syntax
0x10	ISO/IEC 14496-2 Visual
0x11	ISO/IEC 14496-3 Audio with the LATM transport syntax as defined in ISO/IEC 14496-3 / AMD 1
0x12	ISO/IEC 14496-1 SL-packetized stream or FlexMux stream carried in PES packets
0x13	ISO/IEC 14496-1 SL-packetized stream or FlexMux stream carried in ISO/IEC14496_sections.
0x14	ISO/IEC 13818-6 Synchronized Download Protocol
0x15-0x7F	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Reserved
0x80-0xFF	User Private

A private table may be made of several private_sections, all with the same table_id (see Table 2-30).

2.4.4.11 Semantic definition of fields in private section

table_id – This 8-bit field, the value of which identifies the Private Table this section belongs to. Only values defined in Table 2-26 as "user private" may be used.

section_syntax_indicator – This is a 1-bit indicator. When set to '1', it indicates that the private section follows the generic section syntax beyond the private_section_length field. When set to '0', it indicates that the private_data_bytes immediately follow the private_section_length field.

private_indicator – This is a 1-bit user definable flag that shall not be specified by ITU-T | ISO/IEC in the future.

private_section_length – A 12-bit field. It specifies the number of remaining bytes in the private section immediately following the private_section_length field up to the end of the private_section. The value in this field shall not exceed 4093 (0xFFD).

Table 2-30 – Private section

Syntax	No. of bits	Mnemonic
private_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
private_indicator	**1**	**bslbf**
reserved	**2**	**bslbf**
private_section_length	**12**	**uimsbf**
if (section_syntax_indicator == '0') {		
for (i = 0; i < N; i++) {		
private_data_byte	**8**	**bslbf**
}		
}		
else {		
table_id_extension	**16**	**uimsbf**
reserved	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < private_section_length-9; i++) {		
private_data_byte	**8**	**bslbf**
}		
CRC_32	**32**	**rpchof**
}		
}		

private_data_byte – The private_data_byte field is user definable and shall not be specified by ITU-T | ISO/IEC in the future.

table_id_extension – This is a 16-bit field. Its use and value are defined by the user.

version_number – This 5-bit field is the version number of the private_section. The version_number shall be incremented by 1 modulo 32 when a change in the information carried within the private_section occurs. When the current_next_indicator is set to '0', then the version_number shall be that of the next applicable private_section with the same table_id and section_number.

current_next_indicator – A 1-bit field, which when set to '1' indicates that the private_section sent is currently applicable. When the current_next_indicator is set to '1', then the version_number shall be that of the currently applicable private_section. When the bit is set to '0', it indicates that the private_section sent is not yet applicable and shall be the next private_section with the same section_number and table_id to become valid.

section_number – This 8-bit field gives the number of the private_section. The section_number of the first section in a private table shall be 0x00. The section_number shall be incremented by 1 with each additional section in this private table.

last_section_number – This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the private table of which this section is a part.

CRC_32 – This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex A after processing the entire private section.

2.4.4.12 Syntax of the Transport Stream section

ITU-T Rec. H.222.0 | ISO/IEC 13818-1 compliant bitstreams may carry the information defined in Table 2-30-1. ITU-T Rec. H.222.0 | ISO/IEC 13818-1 compliant decoders may decode the information defined in this table.

The Transport Stream Description Table is defined to support the carriage of descriptors as found in 2.6 for an entire Transport Stream. The descriptors shall apply to the entire Transport Stream. This table uses a table_id value of 0x03 as specified in Table 2-26 and is carried in Transport Stream packets whose PID value is 0x0002 as specified in Table 2-3.

Table 2-30-1 – The Transport Stream Description Table

Syntax	No. of bits	Mnemonic
TS_description_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
reserved	**2**	**bslbf**
section_length	**12**	**uimsbf**
reserved	**18**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.13 Semantic definition of fields in the Transport Stream section

table_id – This is an 8-bit field, which shall be set to '0x03' as specified in Table 2-26.

section_length – This is a 12-bit field, the first two bits of which shall be '00'. The remaining 10 bits specify the number of bytes of the section, starting immediately following the section_length field, and including the CRC. The value in this field shall not exceed 1021 (0x3FD).

version_number – This 5-bit field is the version number of the whole Transport Stream Description Table. The version number shall be incremented by 1 modulo 32 whenever the definition of the Transport Stream Description Table changes. When the current_next_indicator is set to '1', then the version_number shall be that of the currently applicable Transport Stream Description Table. When the current_next_indicator is set to '0', then the version_number shall be that of the next applicable Transport Stream Description Table.

current_next_indicator – A 1-bit indicator, which, when set to '1', indicates that the Transport Stream Description Table sent is currently applicable. When the bit is set to '0', it indicates that the table sent is not yet applicable and shall be the next table to become valid.

section_number – This 8-bit field gives the number of this section. The section_number of the first section in the Transport Stream Description Table shall be 0x00. It shall be incremented by 1 with each additional section in the Transport Stream Description Table.

last_section_number – This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the complete Transport Stream Description Table.

CRC_32 – This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex A after processing the entire Transport Stream Description section.

2.5 Program Stream bitstream requirements

2.5.1 Program Stream coding structure and parameters

The ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Stream coding layer allows one program of one or more elementary streams to be combined into a single stream. Data from each elementary stream are multiplexed together with information that allows synchronized presentation of the elementary streams within the program.

A Program Stream consists of one or more elementary streams from one program multiplexed together. Audio and video elementary streams consist of access units.

Elementary Stream data is carried in PES packets. A PES packet consists of a PES packet header followed by packet data. PES packets are inserted into Program Stream packs.

The PES packet header begins with a 32-bit start-code that also identifies the stream (refer to Table 2-18) to which the packet data belongs. The PES packet header may contain just a Presentation Time Stamp (PTS) or both a presentation timestamp and a Decoding Time Stamp (DTS). The PES packet header also contains other optional fields. The packet data contains a variable number of contiguous bytes from one elementary stream.

In a Program Stream, PES packets are organized in packs. A pack commences with a pack header and is followed by zero or more PES packets. The pack header begins with a 32-bit start-code. The pack header is used to store timing and bitrate information.

The Program Stream begins with a system header that optionally may be repeated. The system header carries a summary of the system parameters defined in the stream.

This Recommendation | International Standard does not specify the coded data which may be used as part of conditional access systems. This Recommendation | International Standard does, however, provide mechanisms for program service providers to transport and identify this data for decoder processing, and to correctly reference data which are here specified.

2.5.2 Program Stream system target decoder

The semantics of the Program Stream and the constraints on these semantics require exact definitions of decoding events and the times at which these events occur. The definitions needed are set out in this Specification using a hypothetical decoder known as the Program Stream system target decoder (P-STD).

The P-STD is a conceptual model used to define these terms precisely and to model the decoding process during the construction of Program Streams. The P-STD is defined only for this purpose. Neither the architecture of the P-STD nor the timing described precludes uninterrupted, synchronized playback of Program Streams from a variety of decoders with different architectures or timing schedules.

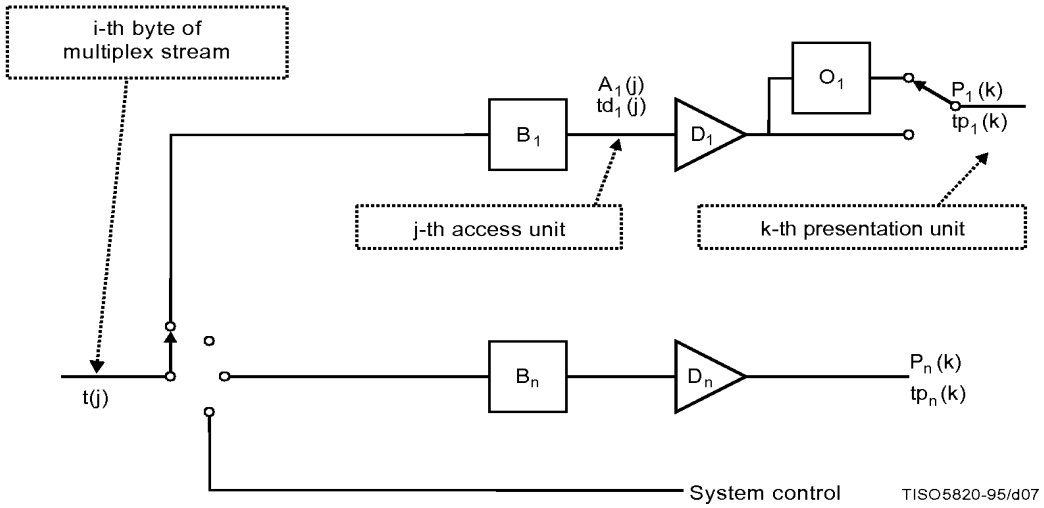


Figure 2-2 – Program Stream system target decoder notation

The following notation is used to describe the Program Stream system target decoder and is partially illustrated in Figure 2-2.

i, i′	are indices to bytes in the Program Stream. The first byte has index 0.
j	is an index to access units in the elementary streams.
k, k′, k″	are indices to presentation units in the elementary streams.
n	is an index to the elementary streams.
t(i)	indicates the time in seconds at which the i-th byte of the Program Stream enters the system target decoder. The value t(0) is an arbitrary constant.
SCR(i)	is the time encoded in the SCR field measured in units of the 27 MHz system clock where i is the byte index of the final byte of the system_clock_reference_base field.
$A_n(j)$	is the j-th access unit in elementary stream n. $A_n(j)$ is indexed in decoding order.
$td_n(j)$	is the decoding time, measured in seconds, in the system target decoder of the j-th access unit in elementary stream n.
$P_n(k)$	is the k-th presentation unit in elementary stream n. $P_n(k)$ is indexed in presentation order.
$tp_n(k)$	is the presentation time, measured in seconds, in the system target decoder of the k-th presentation unit in elementary stream n.

t is time measured in seconds.

$F_n(t)$ is the fullness, measured in bytes, of the system target decoder input buffer for elementary stream n at time t.

B_n the input buffer in the system target decoder for elementary stream n.

BS_n is the size of the system target decoder input buffer, measured in bytes, for elementary stream n.

D_n is the decoder for elementary stream n.

O_n is the reorder buffer for video elementary stream n.

2.5.2.1 System clock frequency

Timing information referenced in P-STD is carried by several data fields defined in this Specification. The fields are defined in 2.5.3.3 and 2.4.3.6. This information is coded as the sampled value of a system clock.

The value of the system clock frequency is measured in Hz and shall meet the following constraints:

– 27 000 000 – 810 <= system_clock_frequency <= 27 000 000 + 810;

– rate of change of system_clock_frequency with time <= 75×10^{-3} Hz/s.

The notation "system_clock_frequency" is used in several places in this Recommendation | International Standard to refer to the frequency of a clock meeting these requirements. For notational convenience, equations in which SCR, PTS, or DTS appear, lead to values of time which are accurate to some integral multiple of (300 ×€2^{33}/system_clock_frequency) seconds. This is due to the encoding of SCR timing information as 33 bits of 1/300 of the system clock frequency plus 9 bits for the remainder, and encoding as 33 bits of the system clock frequency divided by 300 for PTS and DTS.

2.5.2.2 Input to the Program Stream system target decoder

Data from the Program Stream enters the system target decoder. The i-th byte enters at time t(i). The time at which this byte enters the system target decoder can be recovered from the input stream by decoding the input System Clock Reference (SCR) fields and the program_mux_rate field encoded in the pack header. The SCR, as defined in equation 2-18, is coded in two parts: one, in units the period of 1/300 × the system clock frequency, called system_clock_reference_base (see equation 2-19), and one, called system_clock_reference_ext equation (see equation 2-20), in units of the period of the system clock frequency. In the following the values encoded in these fields are denoted by SCR_base(i) and SCR_ext(i). The value encoded in the SCR field indicates time t(i), where i refers to the byte containing the last bit of the system_clock_reference_base field.

Specifically:

$$SCR(i) = SCR\_base(i) \times 300 + SCR\_ext(i) \tag{2-18}$$

where

$$SCR\_base(i) = ((system\_clock\_frequency \times t(i)) \; DIV \; 300) \; \% \; 2^{33} \tag{2-19}$$

$$SCR\_ext(i) = ((system\_clock\_frequency \times t(i)) \; DIV \; 1) \; \% \; 300 \tag{2-20}$$

The input arrival time, t(i), as given in equation 2-21, for all other bytes shall be constructed from SCR(i) and the rate at which data arrives, where the arrival rate within each pack is the value represented in the program_mux_rate field in that pack's header.

$$t(i) = \frac{SCR(i')}{system\_clock\_frequency} + \frac{i - i'}{program\_mux\_rate \times 50} \tag{2-21}$$

where

i′ is the index of the byte containing the last bit of the system_clock_reference_base field in the pack header

i is the index of any byte in the pack, including the pack header

SCR(i′)	is the time encoded in the system clock reference base and extension fields in units of the system clock
program_mux_rate	is a field defined in 2.5.3.3.

After delivery of the last byte of a pack there may be a time interval during which no bytes are delivered to the input of the P-STD.

2.5.2.3 Buffering

The PES packet data from elementary stream n is passed to the input buffer for stream n, B_n. Transfer of byte i from the system target decoder input to B_n is instantaneous, so that byte i enters the buffer for stream n, of size BS_n, at time t(i).

Bytes present in the pack header, system headers, Program Stream Maps, Program Stream Directories, or PES packet headers of the Program Stream such as SCR, DTS, PTS, and packet_length fields, are not delivered to any of the buffers, but may be used to control the system.

The input buffer sizes BS_1 through BS_n are given by the P-STD buffer size parameter in the syntax in equation 2-16 and equation 2-17.

At the decoding time, $td_n(j)$, all data for the access unit that has been in the buffer longest, $A_n(j)$, and any stuffing bytes that immediately precede it that are present in the buffer at the time $td_n(j)$, are removed instantaneously at time $td_n(j)$. The decoding time $td_n(j)$ is specified in the DTS or PTS fields. Decoding times $td_n(j + 1)$, $td_n(j + 2)$, ... of access units without encoded DTS or PTS fields which directly follow access unit j may be derived from information in the elementary stream. Refer to Annex C of ITU-T Rec. H.262 | ISO/IEC 13818-2, ISO/IEC 13818-3, ISO/IEC 11172-2 or ISO/IEC 11172-3. Also refer to 2.7.5. As the access unit is removed from the buffer, it is instantaneously decoded to a presentation unit.

The Program Stream shall be constructed and t(i) shall be chosen so that the input buffers of size BS_1 through BS_n neither overflow nor underflow in the program system target decoder. That is:

$$0 \leq F_n(t) \leq BS_n$$

for all t and n

and

$$F_n(t) = 0$$

instantaneously before t = t(0).

$F_n(t)$ is the instantaneous fullness of P-STD buffer B_n.

An exception to this condition is that the P-STD buffer B_n may underflow when the low_delay flag in the video sequence header is set to '1' (refer to 2.4.2.6) or when trick_mode status is true (refer to 2.4.3.8).

For all Program Streams, the delay caused by system target decoder input buffering shall be less than or equal to one second except for still picture video data and ISO/IEC 14496 streams. The input buffering delay is the difference in time between a byte entering the input buffer and when it is decoded.

Specifically: in the case of no still picture video data and no ISO/IEC 14496 stream the delay is constrained by:

$$tdn(j)\text{-}t(i) <= 1\ sec$$

in the case of still picture video data the delay is constrained by:

$$tdn(j)\text{-}t(i) <= 60\ sec$$

in the case of ISO/IEC 14496 streams the delay is constrained by:

$$tdn(j)\text{-}t(i) <= 10\ sec$$

for all bytes contained in access unit j.

For Program Streams, all bytes of each pack shall enter the P-STD before any byte of a subsequent pack.

When the **low_delay** flag in the video sequence extension is set to '1' (refer to 6.2.2.3 of ITU-T Rec. H.262 | ISO/IEC 13818-2) the VBV buffer may underflow. In this case when the P-STD elementary stream buffer B_n is examined at the time specified by $td_n(j)$, the complete data for the access unit may not be present in the buffer B_n. When this case arises, the buffer shall be re-examined at intervals of two field-periods until the data for the complete access unit is present in the buffer. At this time the entire access unit shall be removed from buffer B_n instantaneously.

VBV buffer underflow is allowed to occur continuously without limit. The P-STD decoder shall remove access unit data from buffer B_n at the earliest time consistent with the paragraph above and any DTS or PTS values encoded in the bitstream. The decoder may be unable to re-establish correct decoding and display times as indicated by DTS and PTS until the VBV buffer underflow situation ceases and a PTS or DTS is found in the bitstream.

2.5.2.4 PES streams

It is possible to construct a stream of data as a contiguous stream of PES packets each containing data of the same elementary stream and with the same stream_id. Such a stream is called a PES stream. The PES-STD model for a PES stream is identical to that for the Program Stream, with the exception that the Elementary Stream Clock Reference (ESCR) is used in place of the SCR, and ES_rate in place of program_mux_rate. The demultiplexor sends data to only one elementary stream buffer.

Buffer sizes BS_n in the PES-STD model are defined as follows:

– *For ITU-T Rec. H.262 | ISO/IEC 13818-2 video:*

$$BS_n = VBV_{max}[\text{profile, level}] + BS_{oh}$$

BS_{oh} = (1/750) seconds × R_{max}[profile, level], where VBV_{max}[profile, level] and R_{max}[profile, level] are the maximum VBV size and bit rate per profile, level, and layer as defined in Tables 8-14 and 8-13, respectively, of ITU-T Rec. H.262 | ISO/IEC 13818-2. BS_{oh} is allocated for PES packet header overhead.

– *For ISO/IEC 11172-2 video:*

$$BS_n = VBV_{max} + BS_{oh}$$

BS_{oh} = (1/750) seconds × R_{max}, where R_{max} and vbv_max refer to the maximum bitrate and maximum vbv_buffer_size for a constrained parameter bitstream in ISO/IEC 11172-2 respectively.

– *For ISO/IEC 11172-3 or ISO/IEC 13818-3 audio:*

$$BS_n = 2848 \text{ bytes}$$

2.5.2.5 Decoding and presentation

Decoding and presentation in the Program Stream system target decoder are the same as defined for the Transport Stream system target decoder in 2.4.2.4, and 2.4.2.5 respectively.

2.5.2.6 P-STD extensions for carriage of ISO/IEC 14496 data

For decoding of ISO/IEC 14496 data carried in a Program Stream the P-STD model is extended. For decoding of individual ISO/IEC 14496 elementary streams in the P-STD see clause 2.11.2. Clause 2.11.3 defines P-STD extensions and parameters for decoding of ISO/IEC 14496 scenes and associated streams.

2.5.3 Specification of the Program Stream syntax and semantics

The following syntax describes a stream of bytes.

2.5.3.1 Program Stream

See Table 2-31.

Table 2-31 – Program Stream

Syntax	No. of bits	Mnemonic
MPEG2_program_stream() {		
do {		
pack()		
} while (nextbits() == pack_start_code)		
MPEG_program_end_code	**32**	**bslbf**
}		

2.5.3.2 Semantic definition of fields in Program Stream

MPEG_program_end_code – The MPEG_program_end_code is the bit string '0000 0000 0000 0000 0000 0001 1011 1001' (0x000001B9). It terminates the Program Stream.

2.5.3.3 Pack layer of Program Stream

See Tables 2-32 and 2.33.

Table 2-32 – Program Stream pack

Syntax	No. of bits	Mnemonic
pack() {		
pack_header()		
while (nextbits() == packet_start_code_prefix) {		
PES_packet()		
}		
}		

Table 2-33 – Program Stream pack header

Syntax	No. of bits	Mnemonic
pack_header() {		
pack_start_code	**32**	**bslbf**
'01'	**2**	**bslbf**
system_clock_reference_base [32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
system_clock_reference_base [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
system_clock_reference_base [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
system_clock_reference_extension	**9**	**uimsbf**
marker_bit	**1**	**bslbf**
program_mux_rate	**22**	**uimsbf**
marker_bit	**1**	**bslbf**
marker_bit	**1**	**bslbf**
reserved	**5**	**bslbf**
pack_stuffing_length	**3**	**uimsbf**
for (i = 0; i < pack_stuffing_length; i++) {		
stuffing_byte	**8**	**bslbf**
}		
if (nextbits() == system_header_start_code) {		
system_header ()		
}		
}		

2.5.3.4 Semantic definition of fields in program stream pack

pack_start_code – The pack_start_code is the bit string '0000 0000 0000 0000 0000 0001 1011 1010' (0x000001BA). It identifies the beginning of a pack.

system_clock_reference_base; system_clock_reference_extension – The system clock reference (SCR) is a 42-bit field coded in two parts. The first part, system_clock_reference_base, is a 33-bit field whose value is given by SCR_base(i) as given in equation 2-19. The second part, system_clock_reference_extension, is a 9-bit field whose value is given by SCR_ext(i), as given in equation 2-20. The SCR indicates the intended time of arrival of the byte containing the last bit of the system_clock_reference_base at the input of the program target decoder.

The frequency of coding requirements for the SCR field are given in 2.7.1.

marker_bit – A marker_bit is a 1-bit field that has the value '1'.

program_mux_rate – This is a 22-bit integer specifying the rate at which the P-STD receives the Program Stream during the pack in which it is included. The value of program_mux_rate is measured in units of 50 bytes/second. The value 0 is forbidden. The value represented in program_mux_rate is used to define the time of arrival of bytes at the input to the P-STD in 2.5.2. The value encoded in the program_mux_rate field may vary from pack to pack in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 program multiplexed stream.

pack_stuffing_length – A 3-bit integer specifying the number of stuffing bytes which follow this field.

stuffing_byte – This is a fixed 8-bit value equal to '1111 1111' that can be inserted by the encoder, for example to meet the requirements of the channel. It is discarded by the decoder. In each pack header no more than 7 stuffing bytes shall be present.

2.5.3.5 System header

See Table 2-34.

Table 2-34 – Program Stream system header

Syntax	No. of bits	Mnemonic
system_header () {		
system_header_start_code	**32**	**bslbf**
header_length	**16**	**uimsbf**
marker_bit	**1**	**bslbf**
rate_bound	**22**	**uimsbf**
marker_bit	**1**	**bslbf**
audio_bound	**6**	**uimsbf**
fixed_flag	**1**	**bslbf**
CSPS_flag	**1**	**bslbf**
system_audio_lock_flag	**1**	**bslbf**
system_video_lock_flag	**1**	**bslbf**
marker_bit	**1**	**bslbf**
video_bound	**5**	**uimsbf**
packet_rate_restriction_flag	**1**	**bslbf**
reserved_bits	**7**	**bslbf**
while (nextbits () == '1') {		
stream_id	**8**	**uimsbf**
'11'	**2**	**bslbf**
P-STD_buffer_bound_scale	**1**	**bslbf**
P-STD_buffer_size_bound	**13**	**uimsbf**
}		
}		

2.5.3.6 Semantic definition of fields in system header

system_header_start_code – The system_header_start_code is the bit string '0000 0000 0000 0000 0000 0001 1011 1011' (0x000001BB). It identifies the beginning of a system header.

header_length – This 16-bit field indicates the length in bytes of the system header following the header_length field. Future extensions of this Specification may extend the system header.

rate_bound – A 22-bit field. The rate_bound is an integer value greater than or equal to the maximum value of the program_mux_rate field coded in any pack of the Program Stream. It may be used by a decoder to assess whether it is capable of decoding the entire stream.

audio_bound – A 6-bit field. The audio_bound is an integer in the inclusive range from 0 to 32 and is set to a value greater than or equal to the maximum number of ISO/IEC 13818-3 and ISO/IEC 11172-3 audio streams in the Program Stream for which the decoding processes are simultaneously active. For the purpose of this subclause, the decoding process of an ISO/IEC 13818-3 or ISO/IEC 11172-3 audio stream is active if the STD buffer is not empty or if a Presentation Unit is being presented in the P-STD model.

fixed_flag – The fixed_flag is a 1-bit flag. When set to '1' fixed bitrate operation is indicated. When set to '0' variable bitrate operation is indicated. During fixed bitrate operation, the value encoded in all system_clock_reference fields in the multiplexed ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream shall adhere to the following linear equation:

$$SCR\_base(i) = ((c1 \times i + c2)\ DIV\ 300)\ \%\ 2^{33} \tag{2-22}$$

$$SCR\_ext(i) = ((c1 \times i + c2)\ DIV\ 300)\ \%\ 300 \tag{2-23}$$

where

c1 is a real-valued constant valid for all i

c2 is a real-valued constant valid for all i

i is the index in the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 multiplexed stream of the byte containing the final bit of any system_clock_reference field in the stream

CSPS_flag – The CSPS_flag is a 1-bit field. If its value is set to '1' the Program Stream meets the constraints defined in 2.7.9.

system_audio_lock_flag – The system_audio_lock_flag is a 1-bit field indicating that there is a specified, constant rational relationship between the audio sampling rate and the system_clock_frequency in the system target decoder. The system_clock_frequency is defined in 2.5.2.1 and the audio sampling rate is specified in ISO/IEC 13818-3. The system_audio_lock_flag may only be set to '1' if, for all presentation units in all audio elementary streams in the Program Stream, the ratio of system_clock_frequency to the actual audio sampling rate, SCASR, is constant and equal to the value indicated in the following table at the nominal sampling rate indicated in the audio stream.

$$SCASR = \frac{system\_clock\_frequency}{audio\_sample\_rate\_in\_the\_P\text{–}STD} \tag{2-24}$$

The notation $\frac{X}{Y}$ denotes real division.

Nominal audio sampling frequency (kHz)	16	32	22,05	44,1	24	48
SCASR	27 000 000 / 16 000	27 000 000 / 32 000	27 000 000 / 22 050	27 000 000 / 44 100	27 000 000 / 24 000	27 000 000 / 48 000

system_video_lock_flag – The system_video_lock_flag is a 1-bit field indicating that there is a specified, constant rational relationship between the video frame rate and the system clock frequency in the system target decoder. Subclause 2.5.2.1 defines system_clock_frequency and the video frame rate is specified in ITU-T Rec. H.262 | ISO/IEC 13818-2. The system_video_lock_flag may only be set to '1' if, for all presentation units in all video elementary streams in the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 program, the ratio of system_clock_frequency to the actual video frame rate, SCFR, is constant and equal to the value indicated in the following table at the nominal frame rate indicated in the video stream.

$$SCFR = \frac{system\_clock\_frequency}{frame\_rate\_in\_the\_P\text{–}STD} \tag{2-25}$$

Nominal frame rate (Hz)	23,976	24	25	29,97	30	50	59,94	60
SCFR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

The values of the ratio SCFR are exact. The actual frame rate differs slightly from the nominal rate in cases where the nominal rate is 23,976, 29,97, or 59,94 frames per second.

video_bound – The video_bound is a 5-bit integer in the inclusive range from 0 to 16 and is set to a value greater than or equal to the maximum number of ITU-T Rec. H.262 | ISO/IEC 13818-2 and ISO/IEC 11172-2 streams in the Program Stream of which the decoding processes are simultaneously active. For the purpose of this subclause, the decoding process of an ITU-T Rec. H.262 | ISO/IEC 13818-2 and ISO/IEC 11172-2 video stream is active if the P-STD buffer is not empty, or if a Presentation Unit is being presented in the P-STD model, or if the reorder buffer is not empty.

packet_rate_restriction_flag – The packet_rate_restriction_flag is a 1-bit flag. If the CSPS flag is set to '1', the packet_rate_restriction_flag indicates which constraint is applicable to the packet rate, as specified in 2.7.9. If the CSPS flag is set to value of '0', then the meaning of the packet_rate_restriction_flag is undefined.

reserved_bits – This 7-bit field is reserved for future use by ISO/IEC. Until otherwise specified by ITU-T | ISO/IEC it shall have the value '111 1111'.

stream_id – The stream_id is an 8-bit field that indicates the coding and elementary stream number of the stream to which the following P-STD_buffer_bound_scale and P-STD_buffer_size_bound fields refer.

If stream_id equals '1011 1000' the P-STD_buffer_bound_scale and P-STD_buffer_size_bound fields following the stream_id refer to all audio streams in the Program Stream.

If stream_id equals '1011 1001' the P-STD_buffer_bound_scale and P-STD_buffer_size_bound fields following the stream_id refer to all video streams in the Program Stream.

If the stream_id takes on any other value it shall be a byte value greater than or equal to '1011 1100' and shall be interpreted as referring to the stream coding and elementary stream number according to Table 2-18.

Each elementary stream present in the Program Stream shall have its P-STD_buffer_bound_scale and P-STD_buffer_size_bound specified exactly once by this mechanism in each system header.

P-STD_buffer_bound_scale – The P-STD_buffer_bound_scale is a 1-bit field that indicates the scaling factor used to interpret the subsequent P-STD_buffer_size_bound field. If the preceding stream_id indicates an audio stream, P-STD_buffer_bound_scale shall have the value '0'. If the preceding stream_id indicates a video stream, P-STD_buffer_bound_scale shall have the value '1'. For all other stream types, the value of the P-STD_buffer_bound_scale may be either '1' or '0'.

P-STD_buffer_size_bound – The P-STD_buffer_size_bound is a 13-bit unsigned integer defining a value greater than or equal to the maximum P-STD input buffer size, BS_n, over all packets for stream n in the Program Stream. If P-STD_buffer_bound_scale has the value '0', then P-STD_buffer_size_bound measures the buffer size bound in units of 128 bytes. If P-STD_buffer_bound_scale has the value '1', then P-STD_buffer_size_bound measures the buffer size bound in units of 1024 bytes. Thus:

$$\text{if (P-STD\_buffer\_bound\_scale} == 0)$$

$$BS_n \leq P - STD\_buffer\_size\_bound \times 128$$

else

$$BS_n \leq P - STD\_buffer\_size\_bound \times 1024$$

2.5.3.7 Packet layer of Program Stream

The packet layer of the Program Stream is defined by the PES packet layer in 2.4.3.6.

2.5.4 Program Stream map

The Program Stream Map (PSM) provides a description of the elementary streams in the Program Stream and their relationship to one another. When carried in a Transport Stream this structure shall not be modified. The PSM is present as a PES packet when the stream_id value is 0xBC (refer to Table 2-18).

NOTE – This syntax differs from the PES packet syntax described in 2.4.3.6.

Definition for the descriptor() fields may be found in 2.6.

2.5.4.1 Syntax of Program Stream map

See Table 2.35.

Table 2-35 – Program Stream map

Syntax	No. of bits	Mnemonic
program_stream_map() {		
packet_start_code_prefix	**24**	**bslbf**
map_stream_id	**8**	**uimsbf**
program_stream_map_length	**16**	**uimsbf**
current_next_indicator	**1**	**bslbf**
reserved	**2**	**bslbf**
program_stream_map_version	**5**	**uimsbf**
reserved	**7**	**bslbf**
marker_bit	**1**	**bslbf**
program_stream_info_length	**16**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
elementary_stream_map_length	**16**	**uimsbf**
for (i = 0; i < N1; i++) {		
stream_type	**8**	**uimsbf**
elementary_stream_id	**8**	**uimsbf**
elementary_stream_info_length	**16**	**uimsbf**
for (i = 0; i < N2; i++) {		
descriptor()		
}		
}		
CRC_32	**32**	**rpchof**
}		

2.5.4.2 Semantic definition of fields in Program Stream map

packet_start_code_prefix – The packet_start_code_prefix is a 24-bit code. Together with the map_stream_id that follows it constitutes a packet start code that identifies the beginning of a packet. The packet_start_code_prefix is the bit string '0000 0000 0000 0000 0000 0001' (0x000001 in hexadecimal).

map_stream_id – This is an 8-bit field whose value shall be 0xBC.

program_stream_map_length – The program_stream_map_length is a 16-bit field indicating the total number of bytes in the program_stream_map immediately following this field. The maximum value of this field is 1018 (0x3FA).

current_next_indicator – This is a 1-bit field, when set to '1' indicates that the Program Stream Map sent is currently applicable. When the bit is set to '0', it indicates that the Program Stream Map sent is not yet applicable and shall be the next table to become valid.

program_stream_map_version – This 5-bit field is the version number of the whole Program Stream Map. The version number shall be incremented by 1 modulo 32 whenever the definition of the Program Stream Map changes. When the current_next_indicator is set to '1', then the program_stream_map_version shall be that of the currently applicable Program Stream Map. When the current_next_indicator is set to '0', then the program_stream_map_version shall be that of the next applicable Program Stream Map.

program_stream_info_length – The program_stream_info_length is a 16-bit field indicating the total length of the descriptors immediately following this field.

marker_bit – A marker_bit is a 1-bit field that has the value '1'.

elementary_stream_map_length – This is a 16-bit field specifying the total length, in bytes, of all elementary stream information in this program stream map. It includes the stream_type, elementary_stream_id, and elementary_stream_info_length fields.

stream_type – This 8-bit field specifies the type of the stream according to Table 2-29. The stream_type field shall only identify elementary streams contained in PES packets. A value of 0x05 is prohibited.

elementary_stream_id – The elementary_stream_id is an 8-bit field indicating the value of the stream_id field in the PES packet headers of PES packets in which this elementary stream is stored.

elementary_stream_info_length – The elementary_stream_info_length is a 16-bit field indicating the length in bytes of the descriptors immediately following this field.

CRC_32 – This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex A after processing the entire program stream map.

2.5.5 Program Stream directory

The directory for an entire stream is made up of all the directory data carried by Program Stream Directory packets identified with the directory_stream_id. The syntax for program_stream_directory packets is defined in Table 2-36.

NOTE 1 – This syntax differs from the PES packet syntax described in 2.4.3.6.

Directory entries may be required to reference I-pictures in a video stream as defined in ITU-T Rec. H.262 | ISO/IEC 13818-2 and ISO/IEC 11172-2. If an I-picture that is referenced in a directory entry is preceded by a sequence header with no intervening picture headers, the directory entry shall reference the first byte of the sequence header. If an I-picture that is referenced in a directory entry is preceded by a group of pictures header with no intervening picture headers and no immediately preceding sequence header, the directory entry shall reference the first byte of the group of pictures header. Any other picture that a directory entry references shall be referenced by the first byte of the picture header.

NOTE 2 – It is recommended that I-pictures immediately following a sequence header should be referenced in directory structures so that the directory contains an entry at every point where the decoder may be reset completely.

Directory references to audio streams as defined in ISO/IEC 13818-3 and ISO/IEC 11172-3 shall be the syncword of the audio frame.

NOTE 3 – It is recommended that the distance between referenced access units not exceed half a second.

Access units shall be referenced in a program_stream_directory packet in the same order that they appear in the bitstream.

2.5.5.1 Syntax of Program Stream directory packet

See Table 2-36.

2.5.5.2 Semantic definition of fields in Program Stream directory

packet_start_code_prefix – The packet_start_code_prefix is a 24-bit code. Together with the stream_id that follows, it constitutes a packet start code that identifies the beginning of a packet. The packet_start_code_prefix is the bit string '0000 0000 0000 0000 0000 0001' (0x000001 in hexadecimal).

directory_stream_id – This 8-bit field shall have a value '1111 1111' (0xFF).

PES_packet_length – The PES_packet_length is a 16-bit field indicating the total number of bytes in the program_stream_directory immediately following this field (refer to Table 2-18).

number_of_access_units – This 15-bit field is the number of access_units that are referenced in this Directory PES packet.

prev_directory_offset – This 45-bit unsigned integer gives the byte address offset of the first byte of the packet start code of the previous Program Stream Directory packet. This address offset is relative to the first byte of the start code of the packet which contains this previous_directory_offset field. The value '0' indicates that there is no previous Program Stream Directory packet.

next_directory_offset – This 45-bit unsigned integer gives the byte address offset of the first byte of the packet start code of the next Program Stream Directory packet. This address offset is relative to the first byte of the start code of the packet which contains this next_directory_offset field. The value '0' indicates that there is no next Program Stream Directory packet.

packet_stream_id – This 8-bit field is the stream_id of the elementary stream that contains the access unit referenced by this directory entry.

PES_header_position_offset_sign – This 1-bit field is the arithmetic sign for the PES_header_position_offset described immediately following. A value of '0' indicates that the PES_header_position_offset is a positive offset. A value of '1' indicates that the PES_header_position_offset is a negative offset.

PES_header_position_offset – This 44-bit unsigned integer gives the byte offset address of the first byte of the PES packet containing the access unit referenced. The offset address is relative to the first byte of the start-code of the packet containing this PES_header_position_offset field. The value '0' indicates that no access unit is referenced.

reference_offset – This 16-bit field is an unsigned integer indicating the position of the first byte of the referenced access unit, measured in bytes relative to the first byte of the PES packet containing the first byte of the referenced access unit.

Table 2-36 – Program Stream directory packet

Syntax	No. of bits	Mnemonic
directory_PES_packet(){		
packet_start_code_prefix	**24**	**bslbf**
directory_stream_id	**8**	**uimsbf**
PES_packet_length	**16**	**uimsbf**
number_of_access_units	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
prev_directory_offset[44..30]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
prev_directory_offset[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
prev_directory_offset[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
next_directory_offset[44..30]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
next_directory_offset[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
next_directory_offset[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
for (i = 0; i < number_of_access_units; i++) {		
packet_stream_id	**8**	**uimsbf**
PES_header_position_offset_sign	**1**	**tcimsbf**
PES_header_position_offset[43..30]	**14**	**uimsbf**
marker_bit	**1**	**bslbf**
PES_header_position_offset[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
PES_header_position_offset[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
reference_offset	**16**	**uimsbf**
marker_bit	**1**	**bslbf**
reserved	**3**	**bslbf**
PTS[32..30]	**3**	**uimsbf**
marker_bit	**1**	**bslbf**
PTS[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
PTS[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
bytes_to_read[22..8]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
bytes_to_read[7..0]	**8**	**uimsbf**
marker_bit	**1**	**bslbf**
intra_coded_indicator	**1**	**bslbf**
coding_parameters_indicator	**2**	**bslbf**
reserved	**4**	**bslbf**
}		
}		

PTS (presentation_time_stamp) – This 33-bit field is the PTS of the access unit that is referenced. The semantics of the coding of the PTS field are as described in 2.4.3.6.

bytes_to_read – This 23-bit unsigned integer is the number of bytes in the Program Stream after the byte indicated by reference_offset that are needed to decode the access unit completely. This value includes any bytes multiplexed at the systems layer including those containing information from other streams.

intra_coded_indicator – This is a 1-bit flag. When set to '1' it indicates that the referenced access unit is not predictively coded. This is independent of other coding parameters that might be needed to decode the access unit. For example, this field shall be coded as '1' for video Intra frames, whereas for 'P' and 'B' frames this bit shall be coded as '0'. For all PES packets containing data which is not from an ITU-T Rec. H.262 | ISO/IEC 13818-2 video stream, this field is undefined (see Table 2-37).

Table 2-37 – Intra_coded indicator

Value	Meaning
0	Not Intra
1	Intra

coding_parameters_indicator – This 2-bit field is used to indicate the location of coding parameters that are needed to decode the access units referenced. For example, this field can be used to determine the location of quantization matrices for video frames.

Table 2-38 – Coding_parameters indicator

Value	Meaning
00	All coding parameters are set to their default values
01	All coding parameters are set in this access unit, at least one of them is not set to a default
10	Some coding parameters are set in this access unit
11	No coding parameters are coded in this access unit

2.6 Program and program element descriptors

Program and program element descriptors are structures which may be used to extend the definitions of programs and program elements. All descriptors have a format which begins with an 8-bit tag value. The tag value is followed by an 8-bit descriptor length and data fields.

2.6.1 Semantic definition of fields in program and program element descriptors

The following semantics apply to the descriptors defined in 2.6.2 through 2.6.34.

descriptor_tag – The descriptor_tag is an 8-bit field which identifies each descriptor.

Table 2-39 provides the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 defined, ITU-T Rec. H.222.0 | ISO/IEC 13818-1 reserved, and user available descriptor tag values. An 'X' in the TS or PS columns indicates the applicability of the descriptor to either the Transport Stream or Program Stream respectively. Note that the meaning of fields in a descriptor may depend on which stream it is used in. Each case is specified in the descriptor semantics below.

descriptor_length – The descriptor_length is an 8-bit field specifying the number of bytes of the descriptor immediately following descriptor_length field.

2.6.2 Video stream descriptor

The video stream descriptor provides basic information which identifies the coding parameters of a video elementary stream as described in ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 11172-2 (see Table 2-40).

Table 2-39 – Program and program element descriptors

descriptor_tag	TS	PS	Identification
0	n/a	n/a	Reserved
1	n/a	n/a	Reserved
2	X	X	video_stream_descriptor
3	X	X	audio_stream_descriptor
4	X	X	hierarchy_descriptor
5	X	X	registration_descriptor
6	X	X	data_stream_alignment_descriptor
7	X	X	target_background_grid_descriptor
8	X	X	Video_window_descriptor
9	X	X	CA_descriptor
10	X	X	ISO_639_language_descriptor
11	X	X	System_clock_descriptor
12	X	X	Multiplex_buffer_utilization_descriptor
13	X	X	Copyright_descriptor
14	X		Maximum_bitrate_descriptor
15	X	X	Private_data_indicator_descriptor
16	X	X	Smoothing_buffer_descriptor
17	X		STD_descriptor
18	X	X	IBP_descriptor
19-26	X		Defined in ISO/IEC 13818-6
27	X	X	MPEG-4_video_descriptor
28	X	X	MPEG-4_audio_descriptor
29	X	X	IOD_descriptor
30	X		SL_descriptor
31	X	X	FMC_descriptor
32	X	X	External_ES_ID_descriptor
33	X	X	MuxCode_descriptor
34	X	X	FmxBufferSize_descriptor
35	X		MultiplexBuffer_descriptor
36-63	n/a	n/a	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Reserved
64-255	n/a	n/a	User Private

Table 2-40 – Video stream descriptor

Syntax	No. of bits	Mnemonic
video_stream_descriptor(){		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
multiple_frame_rate_flag	**1**	**bslbf**
frame_rate_code	**4**	**uimsbf**
MPEG_1_only_flag	**1**	**bslbf**
constrained_parameter_flag	**1**	**bslbf**
still_picture_flag	**1**	**bslbf**
if (MPEG_1_only_flag == '0'){		
profile_and_level_indication	**8**	**uimsbf**
chroma_format	**2**	**uimsbf**
frame_rate_extension_flag	**1**	**bslbf**
reserved	**5**	**bslbf**
}		
}		

2.6.3 Semantic definitions of fields in video stream descriptor

multiple_frame_rate_flag – This 1-bit field when set to '1' indicates that multiple frame rates may be present in the video stream. When set to a value of '0' only a single frame rate is present.

frame_rate_code – This is a 4-bit field as defined in 6.3.3 of ITU-T Rec. H.262 | ISO/IEC 13818-2, except that when the multiple_frame_rate_flag is set to a value of '1' the indication of a particular frame rate also permits certain other frame rates to be present in the video stream, as specified in Table 2-41:

Table 2-41 – Frame rate code

Coded as	Also includes
23,976	
24,0	23,976
25,0	
29,97	23,976
30,0	23,976 24,0 29,97
50,0	25,0
59,94	23,976 29,97
60,0	23,976 24,0 29,97 30,0 59,94

MPEG_1_only_flag – This is a 1-bit field which when set to '1' indicates that the video stream contains only ISO/IEC 11172-2 data. If set to '0' the video stream may contain both ITU-T H.262 | ISO/IEC 13818-2 video data and constrained parameter ISO/IEC 11172-2 video data.

constrained_parameter_flag – This is a 1-bit field which when set to '1' indicates that the video stream shall not contain unconstrained ISO/IEC 11172-2 video data. If this field is set to '0' the video stream may contain both constrained parameters and unconstrained ISO/IEC 11172-2 video streams. If the MPEG_1_only_flag is set to '0', the constrained_parameter_flag shall be set to '1'.

still_picture_flag – This is a 1-bit field, which when set to '1' indicates that the video stream contains only still pictures. If the bit is set to '0' then the video stream may contain either moving or still picture data.

profile_and_level_indication – This bit field is the same manner as the profile_and_level_indication fields in the ITU-T Rec. H.262 | ISO/IEC 13818-2 video stream. The value of this field indicates a profile and level that is equal to or higher than any profile and level in any sequence in the associated video stream. For the purposes of this subclause, an ISO/IEC 11172-2 constrained parameter stream is considered to a be a Main Profile at Low Level stream (MP @ LL).

chroma_format – This 2-bit field is coded in the same manner as the chroma_format fields in the ITU-T Rec. H.262 | ISO/IEC 13818-2 video stream. The value of this field shall be at least equal to or higher than the value of the chroma_format field in any video sequence of the associated video stream. For the purposes of this subclause, an ISO/IEC 11172-2 video stream is considered to have chroma_format field with the value '01', indicating 4:2:0.

frame_rate_extension_flag – This is a 1-bit flag which when set to '1' indicates that either or both the frame_rate_extension_n and the frame_rate_extension_d fields are non-zero in any video sequences of the ITU-T Rec. H.262 | ISO/IEC 13818-2 video stream. For the purposes of this subclause, an ISO/IEC 11172-2 video stream is constrained to have both fields set to zero.

2.6.4 Audio stream descriptor

The audio stream descriptor provides basic information which identifies the coding version of an audio elementary stream as described in ISO/IEC 13818-3 or ISO/IEC 11172-3 (see Table 2-42).

Table 2-42 – Audio stream descriptor

Syntax	No. of bits	Mnemonic
audio_stream_descriptor(){		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
free_format_flag	**1**	**bslbf**
ID	**1**	**bslbf**
layer	**2**	**bslbf**
variable_rate_audio_indicator	**1**	**bslbf**
reserved	**3**	**bslbf**
}		

2.6.5 Semantic definition of fields in audio stream descriptor

free_format_flag – This 1-bit field when set to '1' indicates that the audio stream may contain one or more audio frames with the bitrate_index set to '0000'. If set to '0', then the bitrate_index is not '0000' (refer to 2.4.2.3 of ISO/IEC 13818-3) in any audio frame of the audio stream.

ID – This 1-bit field when set to '1' indicates that the ID field is set to '1' in each audio frame in the audio stream (refer to 2.4.2.3 of ISO/IEC 13818-3).

layer – This 2-bit field is coded in the same manner as the layer field in the ISO/IEC 13818-3 or ISO/IEC 11172-3 audio streams (refer to 2.4.2.3 of ISO/IEC 13818-3). The layer indicated in this field shall be equal to or higher than the highest layer specified in any audio frame of the audio stream.

variable_rate_audio_indicator – This 1-bit flag, when set to '0' indicates that the encoded value of the bit rate field shall not change in consecutive audio frames which are intended to be presented without discontinuity.

2.6.6 Hierarchy descriptor

The hierarchy descriptor provides information to identify the program elements containing components of hierarchically-coded video and audio, and private streams which are multiplexed in multiple streams as described in this Recommendation | International Standard, in ITU-T Rec. H.262 | ISO/IEC 13818-2 and in ISO/IEC 13818-3. (See Table 2-43.)

2.6.7 Semantic definition of fields in hierarchy descriptor

hierarchy_type – The hierarchical relation between the associated hierarchy layer and its hierarchy embedded layer is defined in Table 2-44.

hierarchy_layer_index – The hierarchy_layer_index is a 6-bit field that defines a unique index of the associated program element in a table of coding layer hierarchies. Indices shall be unique within a single program definition.

hierarchy_embedded_layer_index – The hierarchy_embedded_layer_index is a 6-bit field that defines the hierarchy table index of the program element that needs to be accessed before decoding of the elementary stream associated with this hierarchy_descriptor. This field is undefined if the hierarchy_type value is 15 (base layer).

hierarchy_channel – The hierarchy_channel is a 6-bit field that indicates the intended channel number for the associated program element in an ordered set of transmission channels. The most robust transmission channel is defined by the lowest value of this field with respect to the overall transmission hierarchy definition.

NOTE – A given hierarchy_channel may at the same time be assigned to several program elements.

Table 2-43 – Hierarchy descriptor

Syntax	No. of bits	Mnemonic
hierarchy_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
reserved	**4**	**bslbf**
hierarchy_type	**4**	**uimsbf**
reserved	**2**	**bslbf**
hierarchy_layer_index	**6**	**uimsbf**
reserved	**2**	**bslbf**
hierarchy_embedded_layer_index	**6**	**uimsbf**
reserved	**2**	**bslbf**
hierarchy_channel	**6**	**uimsbf**
}		

Table 2-44 – Hierarchy_type field values

Value	Description
0	Reserved
1	ITU-T Rec. H.262 | ISO/IEC 13818-2 Spatial Scalability
2	ITU-T Rec. H.262 | ISO/IEC 13818-2 SNR Scalability
3	ITU-T Rec. H.262 | ISO/IEC 13818-2 Temporal Scalability
4	ITU-T Rec. H.262 | ISO/IEC 13818-2 Data partitioning
5	ISO/IEC 13818-3 Extension bitstream
6	ITU-T Rec.H222.0 | ISO/IEC 13818-1 Private Stream
7	ITU-T Rec. H.262 | ISO/IEC 13818-2 Multi-view Profile
8-14	Reserved
15	Base layer

2.6.8 Registration descriptor

The registration_descriptor provides a method to uniquely and unambiguously identify formats of private data (see Table 2-45).

Table 2-45 – Registration descriptor

Syntax	No. of bits	Identifier
registration_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
format_identifier	**32**	**uimsbf**
for (i = 0; i < N; i++){		
additional_identification_info	**8**	**bslbf**
}		
}		

2.6.9 Semantic definition of fields in registration descriptor

format_identifier – The format_identifier is a 32-bit value obtained from a Registration Authority as designated by ISO/IEC JTC 1/SC 29.

additional_identification_info – The meaning of additional_identification_info bytes, if any, are defined by the assignee of that format_identifier, and once defined they shall not change.

2.6.10 Data stream alignment descriptor

The data stream alignment descriptor describes which type of alignment is present in the associated elementary stream. If the data_alignment_indicator in the PES packet header is set to '1' and the descriptor is present, alignment – as specified in this descriptor – is required (see Table 2-46).

Table 2-46 – Data stream alignment descriptor

Syntax	No. of bits	Mnemonic
data_stream_alignment_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
alignment_type	**8**	**uimsbf**
}		

2.6.11 Semantics of fields in data stream alignment descriptor

alignment_type – Table 2-47 describes the video alignment type when the data_alignment_indicator in the PES packet header has a value of '1'. In each case of alignment_type value the first PES_packet_data_byte following the PES header shall be the first byte of a start code of the type indicated in Table 2-47. At the beginning of a video sequence, the alignment shall occur at the start code of the first sequence header.

NOTE – Specifying alignment type '01' from Table 2-47 does not preclude the alignment from beginning at a GOP or SEQ header.

The definition of access unit for video data is given in 2.1.1.

Table 2-47 – Video stream alignment values

Alignment type	Description
00	Reserved
01	Slice, or video access unit
02	Video access unit
03	GOP, or SEQ
04	SEQ
05-FF	Reserved

Table 2-48 describes the audio alignment type when the data_alignment_indicator in the PES packet header has a value of '1'. In this case the first PES_packet_data_byte following the PES header is the first byte of an audio sync word.

Table 2-48 – Audio stream alignment values

Alignment type	Description
00	Reserved
01	Sync word
02-FF	Reserved

2.6.12 Target background grid descriptor

It is possible to have one or more video streams which, when decoded, are not intended to occupy the full display area (e.g. a monitor). The combination of target_background_grid_descriptor and video_window_descriptors allows the display of these video windows in their desired locations. The target_background_grid_descriptor is used to describe a grid of unit pixels projected on to the display area. The video_window_descriptor is then used to describe, for the associated stream, the location on the grid at which the top left pixel of the display window or display rectangle of the video presentation unit should be displayed. This is represented in Figure 2-3.

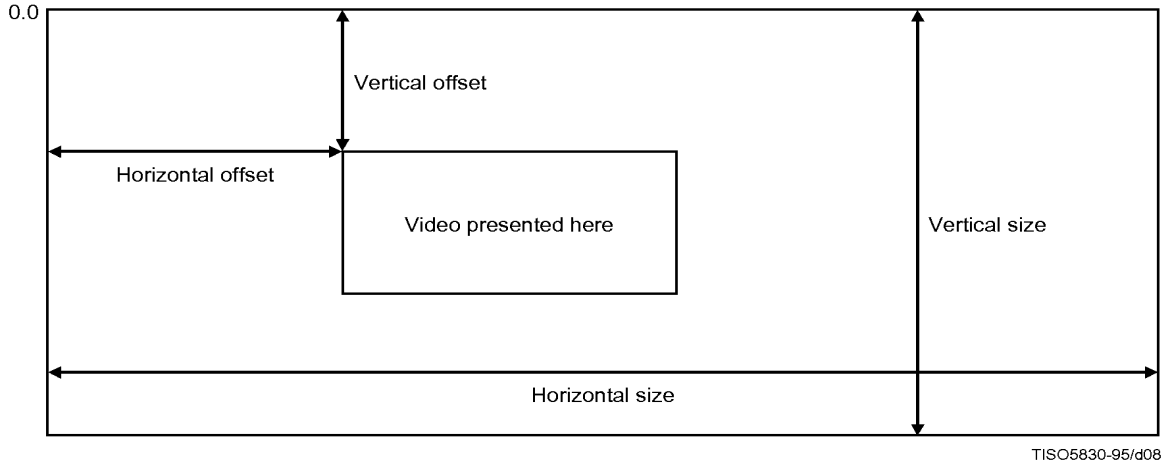


Figure 2-3 – Target background grid descriptor display area

2.6.13 Semantics of fields in target background grid descriptor

horizontal_size – The horizontal size of the target background grid in pixels.

vertical_size – The vertical size of the target background grid in pixels.

aspect_ratio_information – Specifies the sample aspect ratio or display aspect ratio of the target background grid. Aspect_ratio_information is defined in ITU-T Rec. H.262 | ISO/IEC 13818-2 (see Table 2-49).

Table 2-49 – Target background grid descriptor

Syntax	No. of bits	Mnemonic
target_background_grid_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
horizontal_size	**14**	**uimsbf**
vertical_size	**14**	**uimsbf**
aspect_ratio_information	**4**	**uimsbf**
}		

2.6.14 Video window descriptor

The video window descriptor is used to describe the window characteristics of the associated video elementary stream. Its values reference the target background grid descriptor for the same stream. Also see target_background_grid_descriptor in 2.6.12 (see Table 2-50).

Table 2-50 – Video window descriptor

Syntax	No. of bits	Mnemonic
video_window_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
horizontal_offset	**14**	**uimsbf**
vertical_offset	**14**	**uimsbf**
window_priority	**4**	**uimsbf**
}		

2.6.15 Semantic definition of fields in video window descriptor

horizontal_offset – The value indicates the horizontal position of the top left pixel of the current video display window or display rectangle if indicated in the picture display extension on the target background grid for display as defined in the target_background_grid_descriptor. The top left pixel of the video window shall be one of the pixels of the target background grid (refer to Figure 2-3).

vertical_offset – The value indicates the vertical position of the top left pixel of the current video display window or display rectangle if indicated in the picture display extension on the target background grid for display as defined in the target_background_grid_descriptor. The top left pixel of the video window shall be one of the pixels of the target background grid (refer to Figure 2-3).

window_priority – The value indicates how windows overlap. A value of 0 being lowest priority and a value of 15 is the highest priority, i.e. windows with priority 15 are always visible.

2.6.16 Conditional access descriptor

The conditional access descriptor is used to specify both system-wide conditional access management information such as EMMs and elementary stream-specific information such as ECMs. It may be used in both the TS_program_map_section (refer to 2.4.4.8) and the program_stream_map (refer to 2.5.3). If any elementary stream is scrambled, a CA descriptor shall be present for the program containing that elementary stream. If any system-wide conditional access management information exists within a Transport Stream, a CA descriptor shall be present in the conditional access table.

When the CA descriptor is found in the TS_program_map_section (table_id = 0x02), the CA_PID points to packets containing program related access control information, such as ECMs. Its presence as program information indicates applicability to the entire program. In the same case, its presence as extended ES information indicates applicability to the associated program element. Provision is also made for private data.

When the CA descriptor is found in the CA_section (table_id = 0x01), the CA_PID points to packets containing system-wide and/or access control management information, such as EMMs.

The contents of the Transport Stream packets containing conditional access information are privately defined (see Table 2-51).

Table 2-51 – Conditional access descriptor

Syntax	No. of bits	Mnemonic
CA_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
CA_system_ID	**16**	**uimsbf**
reserved	**3**	**bslbf**
CA_PID	**13**	**uimsbf**
for (i = 0; i < N; i++) {		
private_data_byte	**8**	**uimsbf**
}		
}		

2.6.17 Semantic definition of fields in conditional access descriptor

CA_system_ID – This is a 16-bit field indicating the type of CA system applicable for either the associated ECM and/or EMM streams. The coding of this is privately defined and is not specified by ITU-T | ISO/IEC.

CA_PID – This is a 13-bit field indicating the PID of the Transport Stream packets which shall contain either ECM or EMM information for the CA systems as specified with the associated CA_system_ID. The contents (ECM or EMM) of the packets indicated by the CA_PID is determined from the context in which the CA_PID is found, i.e. a TS_program_map_section or the CA table in the Transport Stream, or the stream_id field in the Program Stream.

2.6.18 ISO 639 language descriptor

The language descriptor is used to specify the language of the associated program element (see Table 2-52).

Table 2-52 – ISO 639 language descriptor

Syntax	No. of bits	Mnemonic
ISO_639_language_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
ISO_639_language_code	**24**	**bslbf**
audio_type	**8**	**bslbf**
}		
}		

2.6.19 Semantic definition of fields in ISO 639 language descriptor

ISO_639_language_code – Identifies the language or languages used by the associated program element. The ISO_639_language_code contains a 3-character code as specified by ISO 639, Part 2. Each character is coded into 8 bits according to ISO 8859-1 and inserted in order into this 24-bit field. In the case of multilingual audio streams the sequence of ISO_639_language_code fields shall reflect the content of the audio stream.

audio_type – The audio_type is an 8-bit field which specifies the type of stream defined in Table 2-53.

Table 2-53 – Audio type values

Value	Description
0x00	Undefined
0x01	Clean effects
0x02	Hearing impaired
0x03	Visual impaired commentary
0x04-0xFF	Reserved

clean effects – This field indicates that the referenced program element has no language.

hearing impaired – This field indicates that the referenced program element is prepared for the hearing impaired.

visual_impaired_commentary – This field indicates that the referenced program element is prepared for the visually impaired viewer.

2.6.20 System clock descriptor

This descriptor conveys information about the system clock that was used to generate the timestamps.

If an external clock reference was used, the external_clock_reference_indicator may be set to '1'. The decoder optionally may use the same external reference if it is available.

If the system clock is more accurate than the 30 ppm accuracy required, then the accuracy of the clock can be communicated by encoding it in the clock_accuracy fields. The clock frequency accuracy is:

$$\text{clock\_accuracy\_integer} \times 10^{-\text{clock\_accuracy\_exponent}} \text{ ppm} \qquad (2\text{-}26)$$

If clock_accuracy_integer is set to 0, then the system clock accuracy is 30 ppm. When the external_clock_reference_indicator is set to '1', the clock accuracy pertains to the external reference clock (see Table 2-54).

Table 2-54 – System clock descriptor

Syntax	No. of bits	Mnemonic
system_clock_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
external_clock_reference_indicator	**1**	**bslbf**
reserved	**1**	**bslbf**
clock_accuracy_integer	**6**	**uimsbf**
clock_accuracy_exponent	**3**	**uimsbf**
reserved	**5**	**bslbf**
}		

2.6.21 Semantic definition of fields in system clock descriptor

external_clock_reference_indicator – This is a 1-bit indicator. When set to '1', it indicates that the system clock has been derived from an external frequency reference that may be available at the decoder.

clock_accuracy_integer – This is a 6-bit integer. Together with the clock_accuracy_exponent, it gives the fractional frequency accuracy of the system clock in parts per million.

clock_accuracy_exponent – This is a 3-bit integer. Together with the clock_accuracy_integer, it gives the fractional frequency accuracy of the system clock in parts per million.

2.6.22 Multiplex buffer utilization descriptor

The multiplex buffer utilization descriptor provides bounds on the occupancy of the STD multiplex buffer. This information is intended for devices such as remultiplexers, which may use this information to support a desired re-multiplexing strategy (see Table 2-55).

Table 2-55 – Multiplex buffer utilization descriptor

Syntax	No. of bits	Mnemonic
multiplex_buffer_utilization_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
bound_valid_flag	**1**	**bslbf**
LTW_offset_lower_bound	**15**	**uimsbf**
reserved	**1**	**bslbf**
LTW_offset_upper_bound	**14**	**uimsbf**
}		

2.6.23 Semantic definition of fields in multiplex buffer utilization descriptor

bound_valid_flag – A value of '1' indicates that the LTW_offset_lower_bound and the LTW_offset_upper_bound fields are valid.

LTW_offset_lower_bound – This 15-bit field is defined only if the bound_valid flag has a value of '1'. When defined, this field has the units of (27 MHz / 300) clock periods, as defined for the LTW_offset (refer to 2.4.3.4). The LTW_offset_lower_bound represents the lowest value that any LTW_offset field would have, if that field were coded in every packet of the stream or streams referenced by this descriptor. Actual LTW_offset fields may or may not be coded in the bitstream when the multiplex buffer utilization descriptor is present. This bound is valid until the next occurrence of this descriptor.

LTW_offset_upper_bound – This 15-bit field is defined only if the bound_valid has a value of '1'. When defined, this field has the units of (27 MHz / 300) clock periods, as defined for the LTW_offset (refer to 2.4.3.4). The LTW_offset_upper_bound represents the largest value that any LTW_offset field would have, if that field were coded in every packet of the stream or streams referenced by this descriptor. Actual LTW_offset fields may or may not be coded in the bitstream when the multiplex buffer utilization descriptor is present. This bound is valid until the next occurrence of this descriptor.

2.6.24 Copyright descriptor

The copyright_descriptor provides a method to enable audio-visual works identification. This copyright_descriptor applies to programs or program elements within programs (see Table 2-56).

Table 2-56 – Copyright descriptor

Syntax	No. of bits	Identifier
copyright_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
copyright_identifier	**32**	**uimsbf**
for (i = 0; i < N; i++){		
additional_copyright_info	**8**	**bslbf**
}		
}		

2.6.25 Semantic definition of fields in copyright descriptor

copyright_identifier – This field is a 32-bit value obtained from the Registration Authority.

additional_copyright_info – The meaning of additional_copyright_info bytes, if any, are defined by the assignee of that copyright_identifier, and once defined, they shall not change.

2.6.26 Maximum bitrate descriptor

See Table 2-57.

Table 2-57 – Maximum bitrate descriptor

Syntax	No. of bits	Identifier
maximum_bitrate_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
reserved	**2**	**bslbf**
maximum_bitrate	**22**	**uimsbf**
}		

2.6.27 Semantic definition of fields in maximum bitrate descriptor

maximum_bitrate – The maximum bitrate is coded as a 22-bit positive integer in this field. The value indicates an upper bound of the bitrate, including transport overhead, that will be encountered in this program element or program. The value of maximum_bitrate is expressed in units of 50 bytes/second. The maximum_bitrate_descriptor is included in the Program Map Table (PMT). Its presence as extended program information indicates applicability to the entire program. Its presence as ES information indicates applicability to the associated program element.

2.6.28 Private data indicator descriptor

See Table 2-58.

Table 2-58 – Private data indicator descriptor

Syntax	No. of bits	Identifier
private_data_indicator_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
private_data_indicator	**32**	**uimsbf**
}		

2.6.29 Semantic definition of fields in Private data indicator descriptor

private_data_indicator – The value of the private_data_indicator is private and shall not be defined by ITU-T | ISO/IEC.

2.6.30 Smoothing buffer descriptor

This descriptor is optional and conveys information about the size of a smoothing buffer, SB_n, associated with this descriptor, and the associated leak rate out of that buffer, for the program element(s) that it refers to.

In the case of Transport Streams, bytes of Transport Stream packets of the associated program element(s) present in the Transport Stream are input to a buffer SB_n of size given by sb_size, at the time defined by equation 2-4.

In the case of Program Streams, bytes of all PES packets of the associated elementary streams, are input to a buffer SB_n of size given by sb_size, at the time defined by equation 2-21.

When there is data present in this buffer, bytes are removed from this buffer at a rate defined by sb_leak_rate. The buffer, SB_n shall never overflow. During the continuous existence of a program, the value of the elements of the Smoothing Buffer descriptor of the different program element(s) in the program, shall not change.

The meaning of the smoothing buffer_descriptor is only defined when it is included in the PMT or the Program Stream Map.

If, in the case of a Transport Stream, it is present in the ES info in the Program Map Table, all Transport Stream packets of the PID of that program element enter the smoothing buffer.

If, in the case of a Transport Stream, it is present in the program information, the following Transport Stream packets enter the smoothing buffer:

- all Transport Stream packets of all PIDs listed as elementary_PIDs in the extended program information as well as;
- all Transport Stream packets of the PID which is equal to the PMT_PID of this section;
- all Transport Stream packets of the PCR_PID of the program.

All bytes that enter the associated buffer also exit it.

At any given time there shall be at most one descriptor referring to any individual program element and at most one descriptor referring to the program in its entirety.

Table 2-59 – Smoothing buffer descriptor

Syntax	No. of bits	Mnemonic
smoothing_buffer_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
reserved	**2**	**bslbf**
sb_leak_rate	**22**	**uimsbf**
reserved	**2**	**bslbf**
sb_size	**22**	**uimsbf**
}		

2.6.31 Semantic definition of fields in smoothing buffer descriptor

sb_leak_rate – This 22-bit field is coded as a positive integer. Its contents indicate the value of the leak rate out of the SB_n buffer for the associated elementary stream or other data in units of 400 bits/s.

sb_size – This 22-bit field is coded as a positive integer. Its contents indicate the value of the size of the multiplexing buffer smoothing buffer SB_n for the associated elementary stream or other data in units of 1 byte (see Table 2-59).

2.6.32 STD descriptor

This descriptor is optional and applies only to the T-STD model and to video elementary streams, and is used as specified 2.4.2. This descriptor does not apply to Program Streams (see Table 2-60).

Table 2-60 – STD descriptor

Syntax	No. of bits	Mnemonic
STD_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
reserved	**7**	**bslbf**
leak_valid_flag	**1**	**bslbf**
}		

2.6.33 Semantic definition of fields in STD descriptor

leak_valid_flag – The leak_valid_flag is a 1-bit flag. When set to '1', the transfer of data from the buffer MB_n to the buffer EB_n in the T-STD uses the leak method as defined in 2.4.2.3. If this flag has a value equal to '0', and the vbv_delay fields present in the associated video stream do not have the value 0xFFFF, the transfer of data from the buffer MB_n to the buffer EB_n uses the vbv_delay method as defined in 2.4.2.3.

2.6.34 IBP descriptor

This optional descriptor provides information about some characteristics of the sequence of frame types in the video sequence (see Table 2-61).

Table 2-61 – IBP descriptor

Syntax	No. of bits	Mnemonic
ibp_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
closed_gop_flag	**1**	**uimsbf**
identical_gop_flag	**1**	**uimsbf**
max_gop-length	**14**	**uimsbf**
}		

2.6.35 Semantic definition of fields in IBP descriptor

closed_gop_flag – This 1-bit flag when set to '1' indicates that a group of pictures header is encoded before every I-frame and that the closed_gop flag is set to '1' in all group of pictures headers in the video sequence.

identical_gop_flag – This 1-bit flag when set to '1' indicates that the number of P-frames and B-frames between I-frames, and the picture coding types and sequence of picture types between I-pictures is the same throughout the sequence, except possibly for the pictures up to the second I-picture.

max_gop_length – This 14-bit unsigned integer indicates the maximum number of the coded pictures between any two consecutive I-pictures in the sequence. The value of 0 is forbidden.

2.6.36 MPEG-4 video descriptor

For individual ISO/IEC 14496-2 streams directly carried in PES packets, as defined in subclause 2.11.2, the MPEG-4 video descriptor provides basic information for identifying the coding parameters of such visual elementary streams. The MPEG-4 video descriptor does not apply to ISO/IEC 14496-2 streams encapsulated in SL-packets and in FlexMux packets, as defined in subclause 2.11.3.

Syntax	No. of bits	Mnemonic
MPEG-4_video_descriptor ()		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MPEG-4_visual_profile_and_level	**8**	**uimsbf**
}		

2.6.37 Semantic definition of fields in MPEG-4 video descriptor

MPEG-4_video_profile_and_level – This 8-bit field shall identify the profile and level of the ISO/IEC 14496-2 video stream. This field shall be coded with the same value as the profile_and_level_indication field in the Visual Object Sequence Header in the associated ISO/IEC 14496-2 stream.

2.6.38 MPEG-4 audio descriptor

For individual ISO/IEC 14496-3 streams directly carried in PES packets, as defined in subclause 2.11.2, the MPEG-4 audio descriptor provides basic information for identifying the coding parameters of such audio elementary streams. The MPEG-4 audio descriptor does not apply to ISO/IEC 14496-3 streams encapsulated in SL-packets and in FlexMux packets, as defined in subclause 2.11.3.

Syntax	No. of bits	Mnemonic
MPEG-4_audio_descriptor ()		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MPEG-4_audio_profile_and_level	**8**	**uimsbf**
}		

2.6.39 Semantic definition of fields in MPEG-4 audio descriptor

MPEG-4_audio_profile_and_level – This 8-bit field shall identify the profile and level of the ISO/IEC 14496-3 audio stream corresponding to the Table 2-62.

2.6.40 IOD descriptor

The IOD descriptor encapsulates the InitialObjectDescriptor structure. An initial object descriptor allows access to a set of ISO/IEC 14496 streams by identifying the ES_ID values of the ISO/IEC 14496-1 scene description and object descriptor streams. Both the scene description stream and the object descriptor stream contain further information about the ISO/IEC 14496 streams that are part of the scene. See Annex R for a description of the content access procedure. The InitialObjectDescriptor is specified in subclause 8.6.3 of ISO/IEC 14496-1.

Within a Transport Stream, the IOD descriptor shall be conveyed in the descriptor loop immediately following the program_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the IOD descriptor shall be conveyed in the descriptor loop immediately following the program_stream_info_length field in the Program Stream Map. More than one IOD descriptor may be associated to a program.

NOTE – This Specification does not specify how the IOD_label may be used by higher level service information to uniquely select one of the ISO/IEC 14496 presentations identified by multiple IOD descriptors.

Table 2-62 – MPEG-4_audio_profile_and_level assignment values

Value	Description
0x00-0x0F	Reserved
0x10 0x11 0x12 0x13 0x14-0x17	Main profile, level 1 Main profile, level 2 Main profile, level 3 Main profile, level 4 Reserved
0x18 0x19 0x1A 0x1B 0x1C-0x1F	Scalable Profile, level 1 Scalable Profile, level 2 Scalable Profile, level 3 Scalable Profile, level 4 Reserved
0x20 0x21 0x22-0x27	Speech profile, level 1 Speech profile, level 2 Reserved
0x28 0x29 0x2A 0x2B-0x2F	Synthesis profile, level 1 Synthesis profile, level 2 Synthesis profile, level 3 Reserved
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37	High quality audio profile, level 1 High quality audio profile, level 2 High quality audio profile, level 3 High quality audio profile, level 4 High quality audio profile, level 5 High quality audio profile, level 6 High quality audio profile, level 7 High quality audio profile, level 8
0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F	Low delay audio profile, level 1 Low delay audio profile, level 2 Low delay audio profile, level 3 Low delay audio profile, level 4 Low delay audio profile, level 5 Low delay audio profile, level 6 Low delay audio profile, level 7 Low delay audio profile, level 8
0x40 0x41 0x42 0x43 0x44-0x47	Natural audio profile, level 1 Natural audio profile, level 2 Natural audio profile, level 3 Natural audio profile, level 4 Reserved
0x48 0x49 0x4A 0x4B 0x4C 0x4D 0x4E-0x4F	Mobile audio internetworking profile, level 1 Mobile audio internetworking profile, level 2 Mobile audio internetworking profile, level 3 Mobile audio internetworking profile, level 4 Mobile audio internetworking profile, level 5 Mobile audio internetworking profile, level 6 Reserved
0x50-0xFF	Reserved

Syntax	No. of bits	Mnemonic
IOD_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
Scope_of_IOD_label	**8**	**uimsbf**
IOD_label	**8**	**uimsbf**
InitialObjectDescriptor ()	**8**	**uimsbf**
}		

2.6.41 Semantic definition of fields in IOD descriptor

Scope_of_IOD_label – This 8-bit field specifies the scope of the IOD_label field. A value of 0x10 indicates that the IOD_label is unique within the Program Stream or within the specific program in a Transport Stream in which the IOD descriptor is carried. A value of 0x11 indicates that the IOD_label is unique within the Transport Stream in which the IOD descriptor is carried. All other values of the Scope_of_IOD_label field are reserved.

IOD_label – This 8-bit field specifies the label of the IOD descriptor.

InitialObjectDescriptor () – This structure is defined in subclause 8.6.3.1 of ISO/IEC 14496-1.

2.6.42 SL descriptor

The SL descriptor shall be used when a single ISO/IEC 14496-1 SL-packetized stream is encapsulated in PES packets. The SL descriptor associates the ES_ID of this SL-packetized stream to an elementary_PID in case of a Transport Stream or to an elementary_stream_id in case of a Program Stream. Within a Transport Stream, the SL descriptor shall be conveyed for the corresponding elementary stream in the descriptor loop immediately following the ES_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the SL descriptor shall be conveyed in the descriptor loop immediately following the elementary_stream_info_length field within the Program Stream Map.

NOTE – SL packetized streams may be used in a Program Stream. However, only one stream_id exists for ISO/IEC 14496-1 SL-packetized streams. In order to associate multiple such streams within a Program Stream to an ISO/IEC 14496-1 scene, FlexMux has to be used and signaled appropriately by an FMC descriptor. This limitation does not exist in a Transport Stream where the SL descriptor provides unambiguous mapping between an ISO/IEC 14496-1 ES_ID value and an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 elementary_PID value.

Syntax	No. of bits	Mnemonic
SL_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
ES_ID	**16**	**uimsbf**
}		

2.6.43 Semantic definition of fields in SL descriptor

ES_ID – This 16-bit field shall specify the identifier of an ISO/IEC 14496-1 SL-packetized stream.

2.6.44 FMC descriptor

The FMC descriptor indicates that the ISO/IEC 14496-1 FlexMux tool has been used to multiplex ISO/IEC 14496-1 SL-packetized streams into a FlexMux stream before encapsulation in PES packets or ISO//IEC14496_sections. The FMC descriptor associates FlexMux channels to the ES_ID values of the SL-packetized streams in the FlexMux stream.

An FMC descriptor is required for each program element referenced by an elementary_PID value in a Transport Stream and for each elementary_stream_id in a Program Stream that conveys a FlexMux stream. Within a Transport Stream, the FMC descriptor shall be conveyed for the corresponding elementary stream in the descriptor loop immediately following the ES_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the FMC descriptor shall be conveyed in the descriptor loop immediately following the elementary_stream_info_length field in the Program Stream Map.

For each SL_packetized stream in a FlexMux stream, the FlexMux channel shall be identified by a single entry in the FMC descriptor.

Syntax	No. of bits	Mnemonic
FMC_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i=0; i<descriptor_length; i += 3) {		
ES_ID	**16**	**uimsbf**
FlexMuxChannel	**8**	**uimsbf**
}		
}		

2.6.45 Semantic definition of fields in FMC descriptor

ES_ID – This 16-bit field specifies the identifier of an ISO/IEC 14496-1 SL-packetized stream.

FlexMuxChannel – This 8-bit field specifies the number of the FlexMux channel used for this SL-packetized stream.

2.6.46 External_ES_ID descriptor

The External_ES_ID descriptor assigns an ES_ID, as defined in ISO/IEC 14496-1, to a program element to which no ES_ID value has been assigned by other means. This ES_ID allows reference to a non-ISO/IEC 14496 component in the scene description or, for example, to associate a non-ISO/IEC 14496 component with an IPMP stream.

Within a Transport stream, the assignment of an ES_ID shall be made by conveying an External_ES_ID descriptor for the corresponding elementary stream in the descriptor loop immediately following the ES_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the External_ES_ID descriptor shall be conveyed in the descriptor loop immediately following the elementary_stream_info_length field in the Program Stream Map.

Syntax	No. of bits	Mnemonic
External_ES_ID_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
External_ES_ID	**16**	**uimsbf**
}		

2.6.47 Semantic definition of fields in External_ES_ID descriptor

External_ES_ID – This 16-bit field assigns an ES_ID identifier, as defined in ISO/IEC 14496-1, to a component of a program.

2.6.48 Muxcode descriptor

The Muxcode descriptor conveys MuxCodeTableEntry structures as defined in subclause 11.2.4.3 of ISO/IEC 14496-1. MuxCodeTableEntries configure the MuxCode mode of FlexMux.

One or more Muxcode descriptors may be associated to each elementary_PID or elementary_stream_id, respectively, conveying an ISO/IEC 14496-1 FlexMux stream that utilizes the MuxCode mode. Within a Transport stream, the muxcode descriptor shall be conveyed for the corresponding elementary stream in the descriptor loop immediately following the ES_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the Muxcode descriptor shall be conveyed in the descriptor loop immediately following the elementary_stream_info_length field in the Program Stream Map.

MuxCodeTableEntries may be updated with new versions. In case of such updates, the version_number of each Program Map Table or the program_stream_map_version of each Program Stream Map, respectively, carrying the MuxCode descriptor in their descriptor loop shall be incremented by 1 modulo 32.

Syntax	No. of bits	Mnemonic
Muxcode_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
MuxCodeTableEntry ()		
}		
}		

2.6.49 Semantics of fields in Muxcode descriptor

MuxCodeTableEntry () – This structure is defined in subclause 11.2.4.3 of ISO/IEC 14496-1.

2.6.50 FmxBufferSize descriptor

The FmxBufferSize descriptor conveys the size of the FlexMux buffer (FB) for each SL packetized stream multiplexed in a FlexMux stream.

One FmxBufferSize descriptor shall be associated to each elementary_PID or elementary_stream_id, respectively, conveying an ISO/IEC 14496-1 FlexMux stream. Within a Transport stream, the FmxBufferSize descriptor shall be conveyed for the corresponding elementary stream in the descriptor loop immediately following the ES_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the FmxBufferSize descriptor shall be conveyed in the descriptor loop immediately following the elementary_stream_info_length field within the Program Stream Map.

Syntax	No. of bits	Mnemonic
FmxBufferSize_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
DefaultFlexMuxBufferDescriptor()		
for (i=0; i<descriptor_length; i += 4) {		
FlexMuxBufferDescriptor()		
}		
}		

2.6.51 Semantics of fields in FmxBufferSize descriptor

FlexMuxBufferDescriptor() – This descriptor specifies the FlexMux buffer size for one SL-packetized stream carried within the FlexMux stream. It is defined in subclause 11.2 of ISO/IEC 14496-1.

DefaultFlexMuxBufferDescriptor() – This descriptor specifies the default FlexMux buffer size for this FlexMux stream. It is defined in subclause 11.2 of ISO/IEC 14496-1.

2.6.52 MultiplexBuffer descriptor

The MultiplexBuffer descriptor conveys the size of the multiplex buffer MB_n, as well as the leak rate Rx_n at which data is transferred from transport buffer TB_n into buffer MB_n for a specific ITU-T Rec. H.222.0 | ISO/IEC 13818-1 program element referenced by an elementary_PID value in the Program Map Table.

One MultiplexBuffer descriptor shall be associated to each elementary_PID that contains an ISO/IEC 14496 FlexMux stream or SL-packetized stream, including those containing ISO_IEC_14496_sections. See subclause 2.11.3.9 for the definition of buffers and rates in the T-STD model for decoding of ISO/IEC 14496 content.

The MultiplexBuffer descriptor shall be conveyed in the descriptor loop immediately following the ES_info_length field in the Program Map Table.

Syntax	No. of bits	Mnemonic
MultiplexBuffer_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MB_buffer_size	**24**	**uimsbf**
TB_leak_rate	**24**	**uimsbf**
}		

2.6.53 Semantics of fields in MultiplexBuffer descriptor

MB_buffer_size – This 24-bit field shall specify the size in byte of buffer MB_n of the elementary stream n that is associated with this descriptor.

TB_leak_rate – This 24-bit field shall specify in units of 400 bits per second the rate at which data is transferred from transport buffer TB_n to multiplex buffer MB_n for the elementary stream n that is associated with this descriptor.

2.7 Restrictions on the multiplexed stream semantics

2.7.1 Frequency of coding the system clock reference

The Program Stream shall be constructed such that the time interval between the bytes containing the last bit of system_clock_reference_base fields in successive packs shall be less than or equal to 0.7 s. Thus:

$$\left| t(i) - t(i') \right| \leq 0.7 \text{ s}$$

for all i and i′ where i and i′ are the indexes of the bytes containing the last bit of consecutive system_clock_reference_base fields.

2.7.2 Frequency of coding the program clock reference

The Transport Stream shall be constructed such that the time interval between the bytes containing the last bit of program_clock_reference_base fields in successive occurrences of the PCRs in Transport Stream packets of the PCR_PID for each program shall be less than or equal to 0,1 s. Thus:

$$\left| t(i) - t(i') \right| \leq 0.1 \text{ s}$$

for all i and i′ where i and i′ are the indexes of the bytes containing the last bit of consecutive program_clock_reference_base fields in the Transport Stream packets of the PCR_PID for each program.

There shall be at least two (2) PCRs, from the specified PCR_PID within a Transport Stream, between consecutive PCR discontinuities (refer to 2.4.3.4) to facilitate phase locking and extrapolation of byte delivery times.

2.7.3 Frequency of coding the elementary stream clock reference

The Program Stream and Transport Stream shall be constructed such that if the elementary stream clock reference field is coded in any PES packets containing data of a given elementary stream the time interval in the PES_STD between the bytes containing the last bit of successive ESCR_base fields shall be less than or equal to 0,7 s. In PES Streams the ESCR encoding is required with the same interval. Thus:

$$| t(i) - t(i') | \leq 0.7 \text{ s}$$

for all i and i' where i and i' are the indexes of the bytes containing the last bits of consecutive ESCR_base fields.

NOTE – The coding of elementary stream clock reference fields is optional; they need not be coded. However if they are coded, this constraint applies.

2.7.4 Frequency of presentation timestamp coding

The Program Stream and Transport Stream shall be constructed so that the maximum difference between coded presentation timestamps referring to each elementary video or audio stream is 0,7 s. Thus:

$$| tp_n(k) - tp_n(k'') | \leq 0.7 \text{ s}$$

for all n, k, and k″ satisfying:

- $P_n(k)$ and $P_n(k'')$ are presentation units for which presentation timestamps are coded;
- k and k″ are chosen so that there is no presentation unit, $P_n(k')$ with a coded presentation timestamp and with $k < k' < k''$; and
- No decoding discontinuity exists in elementary stream n between $P_n(k)$ and $P_n(k")$.

In the case of still pictures the 0,7 s constraint does not apply.

2.7.5 Conditional coding of timestamps

For each elementary stream of a Program Stream or Transport Stream, a presentation timestamp (PTS) shall be encoded for the first access unit.

A decoding discontinuity exists at the start of an access unit $A_n(j)$ in an elementary stream n if the decoding time $td_n(j)$ of that access unit is greater than the largest value permissible given the specified tolerance on the system_clock_frequency. For video, except when trick mode status is true or when low_delay flag is '1', this is allowed only at the start of a video sequence. If a decoding discontinuity exists in any elementary video or audio stream in the Transport Stream or Program Stream, then a PTS shall be encoded referring to the first access unit after each decoding discontinuity except when trick mode status is true.

When low_delay is '1' a PTS shall be encoded for the first access unit after an EB_n or B_n underflow.

A PTS may only be present in a ITU-T Rec. H.222.0 | ISO/IEC 13818-1 video or audio elementary stream PES packet header if the first byte of a picture start code or the first byte of an audio access unit is contained in the PES packet.

A decoding_timestamp (DTS) shall appear in a PES packet header if and only if the following two conditions are met:

- a PTS is present in the PES packet header;
- the decoding time differs from the presentation time.

2.7.6 Timing constraints for scalable coding

If an audio sequence is coded using an ISO/IEC 13818-3 extension bitstream, corresponding decoding/presentation units in the two layers shall have identical PTS values.

If a video sequence is coded as a SNR enhancement of another sequence, as specified in 7.8 of ITU-T Rec. H.262 | ISO/IEC 13818-2, the set of presentation times for both sequences shall be the same.

If a video sequence is coded as two partitions, as specified in 7.10 of ITU-T Rec. H.262 | ISO/IEC 13818-2, the set of presentation times for both partitions shall be the same.

If a video sequence is coded as a spatial scalable enhancement of another sequence, as specified in 7.7 of ITU-T Rec. H.262 | ISO/IEC 13818-2, the following shall apply:

- If both sequences have the same frame rate, the set of presentation times for both sequences shall be the same.

  NOTE – that this does not imply that the picture coding type is the same in both layers.

- If the sequences have different frame rates, the set of presentation times shall be such that as many presentation times as possible shall be common to both sequences.
- The picture from which the spatial prediction is made shall be one of the following:
  - the coincident or most recently decoded lower layer picture;
  - the coincident or most recently decoded lower layer picture that is an I- or P-picture;
  - the second most recently decoded lower layer picture that is an I- or P-picture, and provided that the lower layer does not have low_delay set to '1'.

If a video sequence is coded as a temporally scalable enhancement of another sequence, as specified in 7.9 of ITU-T Rec. H.262 | ISO/IEC 13818-2, the following lower layer pictures may be used as the reference. Times are relative to presentation times:

- the coincident or most recently presented lower layer picture;
- the next lower layer picture to be presented.

2.7.7 Frequency of coding P-STD_buffer_size in PES packet headers

In a Program Stream, the P-STD_buffer_scale and P-STD_buffer_size fields shall occur in the first PES packet of each elementary stream and again whenever the value changes. They may also occur in any other PES packet.

2.7.8 Coding of system header in the Program Stream

In a Program Stream, the system header may be present in any pack, immediately following the pack header. The system header shall be present in the first pack of an Program Stream. The values encoded in all the system headers in the Program Stream shall be identical.

2.7.9 Constrained system parameter Program Stream

A Program Stream is a "Constrained System Parameters Stream" (CSPS) if it conforms to the bounds specified in this subclause. Program Streams are not limited to the bounds specified by the CSPS. A CSPS may be identified by means of the CSPS_flag defined in the system header in 2.5.3.5. The CSPS is a subset of all possible Program Streams.

Packet Rate

In the CSPS, the maximum rate at which packets shall arrive at the input to the P-STD is 300 packets per second if the value encoded in the rate_bound field (refer to 2.5.3.6) is less than or equal to 4 500 000 bits/s if the packet_rate_restriction_flag is set to '1', and less than or equal to 2 000 000 bits/s if the packet_rate_restriction_flag is set to '0'. For higher bit rates the CSPS packet rate is bounded by a linear relation to the value encoded in the rate_bound field.

Specifically, for all packs p in the Program Stream when the packet_rate_restriction_flag (refer to 2.5.3.5) is set to a value of '1',

$$NP \leq \left(t(i') - t(i')\right) \times 300 \times \max\left[1, \frac{R_{\max}}{4.5 \times 10^6}\right] \tag{2-27}$$

and if the packet_rate_restriction_flag is set to a value of '0'

$$NP \leq \left(t(i') - t(i')\right) \times 300 \times \max\left[1, \frac{R_{\max}}{2.5 \times 10^6}\right] \tag{2-28}$$

where

$$R_{max} = 8 \times 50 \times rate\_bound \quad \text{bit/s} \tag{2-29}$$

NP is the number of packet_start_code_prefixes and system_header_start_codes between adjacent pack_start_codes or between the last pack_start_code and the MPEG_program_end_code as defined in Table 2-31 and semantics in 2.5.3.2.

t(i) is the time, measured in seconds, encoded in the SCR of pack p.

t(i′) is the time, measured in seconds, encoded in the SCR for pack p + 1, immediately following pack p, or in the case of the final pack in the Program Stream, the time of arrival of the byte containing the last bit of the MPEG_program_end_code.

Decoder Buffer Size

In the case of a CSPS the maximum size of each input buffer in the system target decoder is bounded. Different bounds apply for video elementary streams and audio elementary streams.

In the case of a video elementary stream in a CSPS, the following applies:

BS_n has a size which is equal to the sum of the size of the video buffer verifier (vbv) as specified in ITU-T Rec. H.262 | ISO/IEC 13818-2 and an additional amount of buffering BS_{add}. BS_{add} is specified as:

$$BS_{add} \leq MAX\left[6 \times 1024, R_{vmax} \times 0{,}001\right] \text{ bytes}$$

where R_{vmax} is the maximum video bit rate of the video elementary stream.

In the case of an audio elementary stream in a CSPS, the following applies:

$$BS_n \leq 4096 \text{ bytes}$$

In the case of ISO/IEC 13818-7 ADTS audio elementary stream in a CSPS the following applies to support 8 channels:

$$BS_n \leq 8976 \text{ bytes}$$

2.7.10 Transport Stream

Sample rate locking in Transport Streams

In the Transport Stream there shall be a specified constant rational relationship between the audio sampling rate and the system clock frequency in the system target decoder, and likewise a specified rational relationship between the video frame rate and the system clock frequency. The system_clock_frequency is defined in 2.4.2. The video frame rate is specified in ITU-T Rec. H.262 | ISO/IEC 13818-2 or in ISO/IEC 11172-2. The audio sampling rate is specified in ISO/IEC 13818-3 or in ISO/IEC 11172-3. For all presentation units in all audio elementary streams in the Transport Stream, the ratio of system_clock_frequency to the actual audio sampling rate, SCASR, is constant and equal to the value indicated in the following table at the nominal sampling rate indicated in the audio stream.

$$SCASR = \frac{system\_clock\_frequency}{audio\_sample\_rate\_in\_the\_T\text{-}STD} \tag{2-30}$$

The notation $\frac{X}{Y}$ denotes real division.

Nominal audio sampling frequency (kHz)	16	32	22,05	44,1	24	48
SCASR	27 000 000 / 16 000	27 000 000 / 32 000	27 000 000 / 22 050	27 000 000 / 44 100	27 000 000 / 24 000	27 000 000 / 48 000

For all presentation units in all video elementary streams in the Transport Stream, the ratio of system_clock_frequency to the actual video frame rate, SCFR, is constant and equal to the value indicated in the following table at the nominal frame rate indicated in the video stream.

$$SCFR = \frac{system\_clock\_frequency}{frame\_rate\_in\_the\_T\text{-}STD} \tag{2-31}$$

Nominal frame rate (Hz)	23.976	24	25	29.97	30	50	59.94	60
SCFR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

The values of the SCFR are exact. The actual frame rate differs slightly from the nominal rate in cases where the nominal rate is 23.976, 29.97, or 59.94 frames per second.

2.8 Compatibility with ISO/IEC 11172

The Program Stream of this Recommendation | International Standard is defined to be forward compatible with ISO/IEC 11172-1. Decoders of the Program Stream as defined in this Recommendation | International Standard shall also support decoding of ISO/IEC 11172-1.

2.9 Registration of copyright identifiers

2.9.1 General

Parts 1, 2 and 3 of ISO/IEC 13818 provide support for the management of audiovisual works copyrighting. In ITU-T Rec. H.222.0 | ISO/IEC 13818-1 this is by means of a copyright descriptor, while ITU-T Rec. H.262 | ISO/IEC 13818-2 and ISO/IEC 13818-3 contain fields for identifying copyright holders through syntax fields in the elementary stream syntax. This Recommendation | International Standard presents the method of obtaining and registering copyright identifiers in ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

ITU-T Rec. H.222.0 | ISO/IEC 13818-1 specifies a unique 32-bit copyright_identifier which is a work type code identifier (such as ISBN, ISSN, ISRC, etc.) carried in the copyright descriptor. The copyright_identifier enables identification of a wide number of Copyright Registration Authorities. Each Copyright Registration Authority may specify a syntax and semantic for identifying the audiovisual works or other copyrighted works within that particular copyright organization through appropriate use of the variable length additional_copyright_info field which contains the copyright number.

In the following subclause and Annexes L, M and N, the benefits and responsibilities of all parties to the registration of copyright_identifier are outlined.

2.9.2 Implementation of a Registration Authority (RA)

ISO/IEC JTC 1 shall call for nominations for an international organization which will serve as the Registration Authority for the **copyright_identifier** as defined in 2.6.24 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1. The selected organization shall serve as the Registration Authority. The so-named Registration Authority shall execute its duties in compliance with Annex H/JTC 1 Directives. The registered copyright_identifier is hereafter referred to as the Registered Identifier (RID).

Upon selection of the Registration Authority, JTC 1 shall require the creation of a Registration Management Group (RMG) which will review appeals filed by organizations whose request for a RID to be used in conjunction with ITU-T Rec. H.222.0 | ISO/IEC 13818-1 has been denied by the Registration Authority.

Annexes L, M and N provide information on the procedure for registering a unique copyright identifier.

2.10 Registration of private data format

The registration descriptor of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 is provided by this text in order to enable users of this Specification to unambiguously carry data when its format is not recognized by this Specification. This provision will permit this Specification to carry all types of data while providing for a method of unambiguous identification of the characteristics of the underlying private data.

2.10.1 General

In the following subclause and Annexes O and P, the benefits and responsibilities of all parties to the registration of private data format are outlined.

2.10.2 Implementation of a Registration Authority (RA)

ISO/IEC JTC 1/SC 29 shall call for nominations from member bodies of ISO or National Committees of IEC which will serve as the Registration Authority for the **format_identifier** as defined in 2.6.8 and 2.6.9. The selected organization shall serve as the Registration Authority. The so-named Registration Authority shall execute its duties in compliance with Annex H/JTC 1 Directives. The registered private data format_identifier is hereafter referred to as the Registered Identifier (RID).

Upon selection of the Registration Authority, JTC 1 shall require the creation of a Registration Management Group (RMG) which will review appeals filed by organizations whose request for an RID to be used in conjunction with this Specification has been denied by the Registration Authority.

Annexes O and P provide information on the procedures for registering a unique format identifier.

2.11 Carriage of ISO/IEC 14496 data

2.11.1 Introduction

An ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream may carry individual ISO/IEC 14496-2 and 14496-3 elementary streams as well as ISO/IEC 14496-1 audiovisual scenes with its associated streams. Typically, the ISO/IEC 14496 streams will be elements of an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 program, as defined by the PMT in a Transport Stream and the PSM in a Program Stream.

For the carriage of ISO/IEC 14496 data in Transport Streams and Program Streams distinction is made between individual elementary streams and an ISO/IEC 14496-1 audiovisual scene with its associated streams.

For carriage of individual ISO/IEC 14496-2 and 14496-3 elementary streams only system tools from ITU-T Rec. H.222.0 | ISO/IEC 13818-1 are used, as defined in subclause 2.11.2;

For carriage of an audiovisual ISO/IEC 14496-1 scene and associated ISO/IEC 14496 elementary streams, contained in ISO/IEC 14496-1 SL_packetized streams or FlexMux streams, tools from both ITU-T Rec. H.222.0 | ISO/IEC 13818-1 and from ISO/IEC 14496-1 are used, as defined in subclause 2.11.3.

2.11.2 Carriage of individual ISO/IEC 14496-2 and 14496-3 Elementary Streams in PES packets

2.11.2.1 Introduction

Individual ISO/IEC 14496-2 and 14496-3 elementary streams may be carried in PES packets as PES_packet_data_bytes. For PES packetization no specific data alignment constraints apply. For synchronisation PTSs and, when appropriate, DTSs are encoded in the header of the PES packet that carries the ISO/IEC 14496 elementary stream data; for PTS and DTS encoding the same constraints apply as for ISO/IEC 13818 elementary streams. See Table 2-63 for an overview of how to carry individual ISO/IEC 14496 streams within an ISO/IEC ITU-T Rec. H.222.0 | 13818-1 stream.

Table 2-63 – Carriage of individual ISO/IEC 14496 streams in ITU-T Rec. H.222.0 | ISO/IEC 13818-1

ISO/IEC 14496-2 visual	Carriage in PES packets	Stream_type = 0x10	Stream_id = '1110 xxxx'
ISO/IEC 14496-3 audio	Carriage in PES packets	Stream_type = 0x11	Stream_id = '110x xxxx'

If a PTS or DTS is present in the PES packet header it shall refer to the visual object that follows either the first VOP start code or the first still texture object startcode that commences in the PES packet. Each ISO/IEC 14496-2 video stream carried by ITU-T Rec. H.222.0 | ISO/IEC 13818-1 shall contain the information required to decode the ISO/IEC 14496-2 video stream; consequently the stream shall contain Visual Object Sequence Headers, Visual Object Headers and Video Object Layer Headers.

In case of an ISO/IEC 14496-3 elementary stream, before PES packetization the elementary stream data shall be first encapsulated in the LATM transport syntax defined in ISO/IEC 14496-3 / AMD 1. If a PTS is present in the PES packet header it shall refer to the first audio frame that follows the first syncword that commences in the payload of the PES packet.

Carriage of individual ISO/IEC 14496 elementary streams in PES packets shall be identified by appropriate stream_id and stream_type values, indicating the use of ISO/IEC 14496-2 Visual or 14496-3 Audio. In addition, such carriage shall be signaled by the MPEG-4_video descriptor or MPEG-4_audio descriptor, respectively. These descriptors shall be conveyed in the descriptor loop for the respective elementary stream entry in the Program Map Table in case of a Transport Stream or in the Program Stream Map, when present, in case of a Program Stream. ITU-T Rec. H.222.0 | ISO/IEC 13818-1 does not specify presentation of ISO/IEC 14496 elementary streams in the context of a program.

2.11.2.2 STD extensions for individual ISO/IEC 14496 elementary streams

The T-STD model includes a transport buffer TB_n and a multiplex buffer B_n prior to decoding of each individual ISO/IEC 14496 elementary stream n. Note that in the T-STD the single multiplex buffer B_n is also applied for ISO/IEC 14496-2 video, as indicated in Figure 2-4, instead of the approach with two buffers MB_n and EB_n used for ISO/IEC 13818-2 video in the T-STD. For buffers TB_n and B_n and the rate Rx_n between TB_n and B_n the following constraints apply.

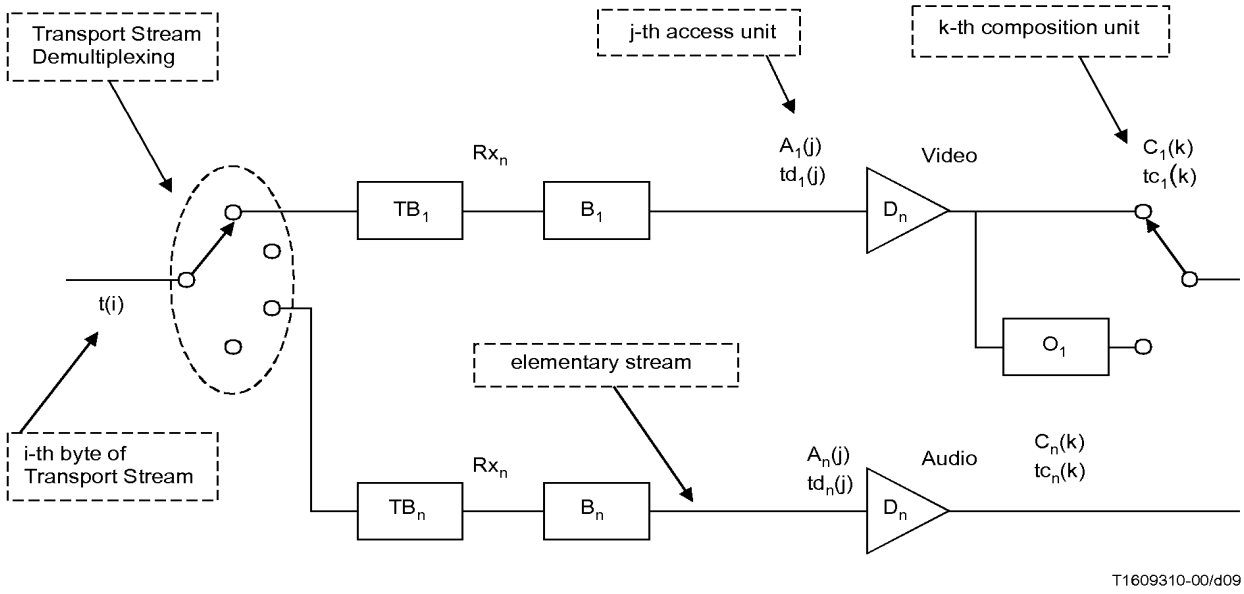


Figure 2-4 – T-STD model extensions for individual ISO/IEC 14496 elementary streams

In case of carriage of an ISO/IEC 14496-2 stream:

Size BS_n of Buffer B_n:

$$BS_n = BS_{mux} + BS_{oh} + VBV_{max}[profile,level]$$

where:

BS_{oh}, packet overhead buffering, is defined as:

$BS_{oh} = (1/750)$ seconds $\times \max\{R_{max}[profile,level], 2\ 000\ 000$ bit/second$\}$

and

BS_{mux}, additional multiplex buffering, is defined as:

$BS_{mux} = 0.004$ seconds $\times$ max{R_{max}[profile,level], 2 000 000 bit/second}

Rate Rx_n:

$$Rx_n = 1.2 \times R_{max}[profile,level]$$

Where

VBV_{max}[profile,level] and R_{max}[profile,level] are defined in ISO/IEC 14496-2 for each profile and level. For profiles and levels for which no VBV_{max} value is specified, the size of B_n and the rate Rx_n are user defined.

In case of carriage of an ISO/IEC 14496-3 stream:

Size BS_n of Buffer B_n for ISO/IEC 14496-3 AAC audio as defined in this Specification,

$$\text{else } BS_n = BS_{mux} + BS_{dec} + BS_{oh} = 3584 \text{ bytes}$$

In this case the size of the access unit decoding buffer BS_{dec}, and the PES packet overhead buffer BS_{oh} are constrained by:

$$BS_{dec} + BS_{oh} \leq 2848 \text{ bytes}$$

A portion (736 bytes) of the 3584 byte buffer is allocated for buffering to allow multiplexing. The rest, 2848 bytes, are shared for access unit buffering BS_{dec}, BS_{oh} and additional multiplexing.

Rate Rx_n for ISO/IEC 14496-3 AAC audio as defined in this Specification,

$$\text{else } Rx_n = 2\ 000\ 000 \text{ bit/sec}$$

The P-STD model includes a multiplex buffer B_n prior to decoding of each individual ISO/IEC 14496 elementary stream n. The size BS_n of buffer B_n in the P-STD is defined by the P-STD_buffer_size field in the PES packet header.

2.11.3 Carriage of audiovisual ISO/IEC 14496-1 scenes and associated ISO/IEC 14496 streams

2.11.3.1 Introduction

This clause describes the encapsulation and signaling when an audiovisual scene represented by ISO/IEC 14496 data is carried in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Stream or Transport Stream. ISO/IEC 14496 content consists of the initial object descriptor and a variable number of streams such as object descriptor streams, scene description streams (carrying either BIFS-Command or BIFS-Anim access units), IPMP streams, OCI streams and audio-visual streams. Each of the ISO/IEC 14496 streams shall be contained in an SL-packetized stream and may optionally be multiplexed into a FlexMux stream, both defined in ISO/IEC 14496-1. For carriage in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Stream or Transport Stream, these SL-packetized streams and FlexMux streams shall contain encoded Object Clock Reference (OCR) and FlexMux Clock Reference (FCR) fields as specified in 2.11.3.4 and in 2.11.3.5, respectively. The SL-packetized streams or FlexMux streams are then encapsulated either in PES packets or in ISO_IEC_14496_sections prior to Transport Stream packetization and multiplexing or Program Stream multiplexing. ISO_IEC_14496_sections are built on the long format of H.222.0 | ISO/IEC 13818-1 sections.

2.11.3.2 Assignment of ES_ID values

An ISO/IEC 14496-1 scene carried over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream may associate a number of ISO/IEC 14496, ISO/IEC 13818 and other streams by the use of the ES_ID parameter. The scene and the associated streams may be carried over the same ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, but a scene may also reference streams carried elsewhere, for example over an IP network. How to identify such other means is not defined in this Specification.

ISO/IEC 14496-1 defines name scoping rules for identifiers. These rules allow the same ES_ID value to be used for two different streams within ISO/IEC 14496 content. When one or multiple ISO/IEC 14496-1 scenes are carried in a ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program, duplicate ES_ID values shall not occur within the program such that each ISO/IEC 14496 SL-packetized stream or ISO/IEC 14496-1 Flexmux channel has a unique ES_ID value in the program.

2.11.3.3 Timing of ISO/IEC 14496 scenes and associated streams

When carried over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, the object time base of each ISO/IEC 14496 stream shall be locked to the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 STC, that is:

$$\text{If } X(t) = f_{stc}(t) / f_{object}(t),$$

then the value of X(t) shall be constant at any time t.

Where

$f_{stc}(t)$ denotes the intended frequency of the STC at time t, i.e. 27 000 000 Hz

$f_{object}(t)$ denotes the frequency of the object time base at time t

The object time base of ISO/IEC 14496 streams carried over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream is conveyed as follows:

- The object time base of an SL-packetized stream carried in PES packets without the use of the FlexMux shall be conveyed by coded OCRs in the SL packet header of that stream. See subclause 2.11.3.4.
- The object time base of SL-packetized streams carried in PES packets within a FlexMux stream shall be conveyed by FCRs in that FlexMux stream. See subclause 2.11.3.5. Consequently, all ISO/IEC 14496 streams contained within the same FlexMux stream share the same object time base.
- The object time base of an SL-packetized stream carried in sections shall be conveyed by another ISO/IEC 14496 stream within the Transport Stream or Program Stream as indicated by the OCR_ES_ID field in the ES descriptor for that stream.

The following constraints shall apply for encoding of OCRs and FCRs in SL-packetized streams and FlexMux streams carried over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream:

- The OCRs and FCRs in each SL-packetized stream and each FlexMux stream associated to the same scene shall have the same resolution.
- The resolution of OCRs and FCRs for a scene, f_{cr}, shall have a value smaller than or equal to 90 000 Hz.
- The ratio $(f_{stc}(t) / 300) / f_{cr}$, shall be an integer value larger than or equal to one. Consequently the resolution of the OCR and FCR syntax elements may only take values such as 90 000 Hz, 45 000 Hz, 30 000 Hz, 22 500 Hz, 18 000 Hz, etc.

Within the above constraints and the ISO/IEC 14496-1 constraint that the resolution f_{cr} shall represent an integer number of cycles per second, f_{cr} can be selected as appropriate for the scene.

The ISO/IEC 14496 time stamps coded in the SL packet header shall refer to instants of the object time base of the stream carried in the SL packet. The resolution of each such time stamp shall be of a factor 2^k smaller than the resolution of the OCRs or FCRs associated to the stream, with k a positive integer larger than or equal to zero. To achieve the same wrap around, the length of the time stamp fields, TimeStampLength, shall be k bit smaller than the length of the OCR or FCR field, OCRLength and FCRLength, respectively. Hence for each stream the following conditions shall apply for encoding of time stamps:

- TimeStampResolution = (OCRResolution or FCRResolution respectively) / 2^k, with k a positive integer larger than or equal to zero. ISO/IEC 14496-1 requires TimeStampResolution to represent an integer number of cycles per second.
- TimeStampLength = OCRLength or FCRLength respectively – k.

The relationship between a value of the STC and the corresponding value of the object time base of a stream is established by associating PTS fields in PES packet headers with the OCR or FCR in SL packet headers and FlexMux Stream packets, respectively, as specified in subclauses 2.11.3.6 and 2.11.3.7.

2.11.3.4 Delivery timing of SL-packetized streams

To carry ISO/IEC 14496 content in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, ISO/IEC 14496-1 SL-packetized streams are used. In each SL-packetized stream carried in a PES packet without the use of FlexMux, the objectClockReference field shall be encoded as follows:

1) An objectClockReference (OCR) field shall be present in the first SL packet header of a SL-packetized stream.

2) The SL-packetized stream shall be constructed such that the time interval between the bytes containing the last bit of successive OCR fields shall be less than or equal to 0,7 second. Thus:

$$| t(i'') - t(i') | <= 0,7 \text{ s}$$

for all i' and i'' where i' and i'' are the indexes of the bytes containing the last bit of consecutive OCR fields in the FlexMux stream.

If an objectClockReference is encoded in an SL packet header, also the instantBitrate field shall be coded.

2.11.3.5 Delivery timing of FlexMux streams

Next to SL-packetized streams also the ISO/IEC 14496-1 FlexMux tool may be used to carry ISO/IEC 14496 content in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 streams. The payload of FlexMux packets shall consist of SL packets as specified in ISO/IEC 14496-1. In each FlexMux stream carried in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream the fmxClockReference field shall be encoded as follows:

1) An fmxClockReference (FCR) field shall be present in the first FlexMux packet of a FlexMux stream.

2) The FlexMux stream shall be constructed such that the time interval between the bytes containing the last bit of successive FCR fields shall be less than or equal to 0,7 second. Thus:

$$| t(i'') - t(i') | <= 0,7 \text{ s}$$

for all i' and i'' where i' and i'' are the indexes of the bytes containing the last bit of consecutive FCR fields in the FlexMux stream.

3) All ISO/IEC 14496 time stamps within the SL-packetized streams carried within a FlexMux stream shall refer to instants of the object time base conveyed by the FCR fields in the FlexMux stream. The SL-packetized streams carried in FlexMux packets need not carry OCR fields. If OCR fields are present, they may be ignored.

2.11.3.6 Carriage of SL-packetized streams in PES packets

A single ISO/IEC 14496-1 SL-packetized stream may be mapped into a single PES stream. One and only one SL packet from an SL-packetized stream shall constitute the payload of one PES packet. PES packets that carry an SL-packetized stream shall be identified by stream_id = 0xFA in the PES packet header.

When an OCR field is coded in the SL packet header, a PTS shall be encoded in the header of the PES packet that carries such SL packet header. This PTS shall be encoded with the 33 bit value of the 90 kHz portion of the STC that corresponds to the value of the object time base at the instant in time indicated by the OCR.

The ES_ID associated to the SL-packetized stream shall be signaled by an SL descriptor as specified in 2.6.46.

2.11.3.7 Carriage of FlexMux streams in PES packets

PES packets with a payload consisting of FlexMux packets shall be identified by stream_id = 0xFB in the PES packet header. An integer number of FlexMux packets shall constitute the payload of one PES packet, i.e., the payload of a PES packet carrying a FlexMux stream shall start with a FlexMux packet header and shall end with the last byte of a FlexMux packet.

If an fmxClockReference (FCR) field is encoded in one of the FlexMux packets contained in a PES packet, then a PTS shall be encoded in the header of the PES packet that contains such FlexMux packet. This PTS shall be encoded with the 33 bit value of the 90 kHz portion of the STC that corresponds to the value of the object time base of the FlexMux stream at the instant in time indicated by the FCR. In case multiple FlexMux packets with an encoded FCR field are contained in a PES packet, the PTS shall correspond to the time indicated by the FCR in the first such FlexMux packet encountered in the payload of the PES packet.

The ES_IDs associated to each SL-packetized stream conveyed in the FlexMux stream shall be signaled by an FMC descriptor as specified in 2.6.38.

2.11.3.8 Carriage of SL packets and FlexMux packets in sections

For transport of ISO/IEC 14496 content in sections, ISO_IEC_14496_sections are defined. Only SL-packetized object descriptor streams and scene description streams shall use ISO_IEC_14496_sections. A single ISO_IEC_14496_section shall contain either an entire SL packet of an SL-packetized stream or an integer number of FlexMux packets each carrying an SL packet of the same ISO/IEC 14496-1 elementary stream.

Table 2-64 shows the syntax of ISO_IEC_14496_sections defined to convey ISO/IEC 14496-1 elementary streams, qualified by the table_id as either object descriptor or scene description stream data. Object descriptor stream data consists of an Object Descriptor Table that comprises a number of object descriptors. The Object Descriptor Table may be transmitted in multiple ISO_IEC_14496_sections. Scene description data consists of a Scene Description Table that may comprise a number of BIFS commands. The Scene Description Table may be transmitted in multiple ISO_IEC_14496_sections. It is not required that a complete table be received in order to process its payload. However, the payload of sections shall be processed in the correct order, as indicated by the value of the section_number field in the ISO_IEC_14496_section header bytes.

Table 2-64 – Section syntax for transport of ISO/IEC14496 streams

Syntax	No. of bits	Mnemonic
ISO_IEC_14496_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
private_indicator	**1**	**bslbf**
reserved	**2**	**bslbf**
ISO_IEC_14496_section_length	**12**	**uimsbf**
table_id_extension	**16**	**uimsbf**
reserved	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
if (PMT_has_SL_descriptor(current_PID)) {		
SL_Packet()		
}		
else if (PMT_has_FMC_descriptor(current_PID)) {		
for (i=1; i<N; i++)		
FlexMuxPacket()		
}		
else {		
for (i=1; i<N; i++)		
reserved	**8**	**bslbf**
}		
CRC_32	**32**	**rpchof**
}		

table_id – This 8-bit field shall be set to '0x04' or '0x05' in case of an ISO_IEC_14496_section. A value of '0x04' indicates an ISO_IEC_14496_scene_description_section that carries an ISO/IEC 14496-1 scene description stream. A value of '0x05' indicates an ISO_IEC_14496_object_descriptor_section that carries an ISO/IEC 14496-1 object descriptor stream.

section_syntax_indicator – This 1-bit field shall be set to '1'.

private_indicator – This 1-bit field shall not be specified by this Specification.

ISO_IEC_14496_section_length – This 12-bit field shall specify the number of remaining bytes in the section immediately following the ISO_IEC_14496_section_length field up to the end of the ISO_IEC_14496_section. The value of this field shall not exceed 4093 (0xFFD).

table_id_extension – This 16-bit field shall not be specified by this Specification; its use and value are define by the user.

version_number – This 5-bit field shall represent the version number of the Object Descriptor Table or Scene Description Table respectively. The version number shall be incremented by 1 modulo 32 with each new version of the table. Version control is at the discretion of the application.

current_next_indicator – This 1-bit field shall be set to 1.

section_number – This 8-bit field shall represent the number of the ISO_IEC_14496_section. The section_number field of the first ISO_IEC_14496_section of the Object Descriptor Table or the Scene Description Table shall have a value equal to 0x00. The value of section_number shall be incremented by 1 with each additional section in the table.

last_section_number – This 8-bit field shall specify the number of the last section of the Object Descriptor Table or Scene Description Table of which this section is a part.

PMT_has_SL_descriptor(current_PID) – a pseudo function that shall be true if an SL descriptor is contained in the descriptor loop in the Program Map Table for the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 program element that conveys this ISO_IEC_14496_section.

SL_Packet() – a sync layer packet as specified in subclause 10.2.2 of ISO/IEC 14496-1.

PMT_has_FMC_descriptor(current_PID) – a pseudo function that shall be true if an FMC descriptor is contained in the descriptor loop in the Program Map Table for the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 program element that conveys this ISO_IEC_14496_section.

FlexMuxPacket() – a FlexMux packet as specified in subclause 11.2.4 of ISO/IEC 14496-1.

CRC_32 – This 32-bit field shall contain the CRC value that gives a zero output of the registers in the decoder defined in Annex A of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 after processing the entire ISO_IEC_14496_section.

2.11.3.9 T-STD extensions

2.11.3.9.1 T-STD Model for 14496 content

Figure 2-5 shows extensions of the Transport System Target Decoder for delivery of ISO/IEC 14496 program elements encapsulated in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Transport Streams.

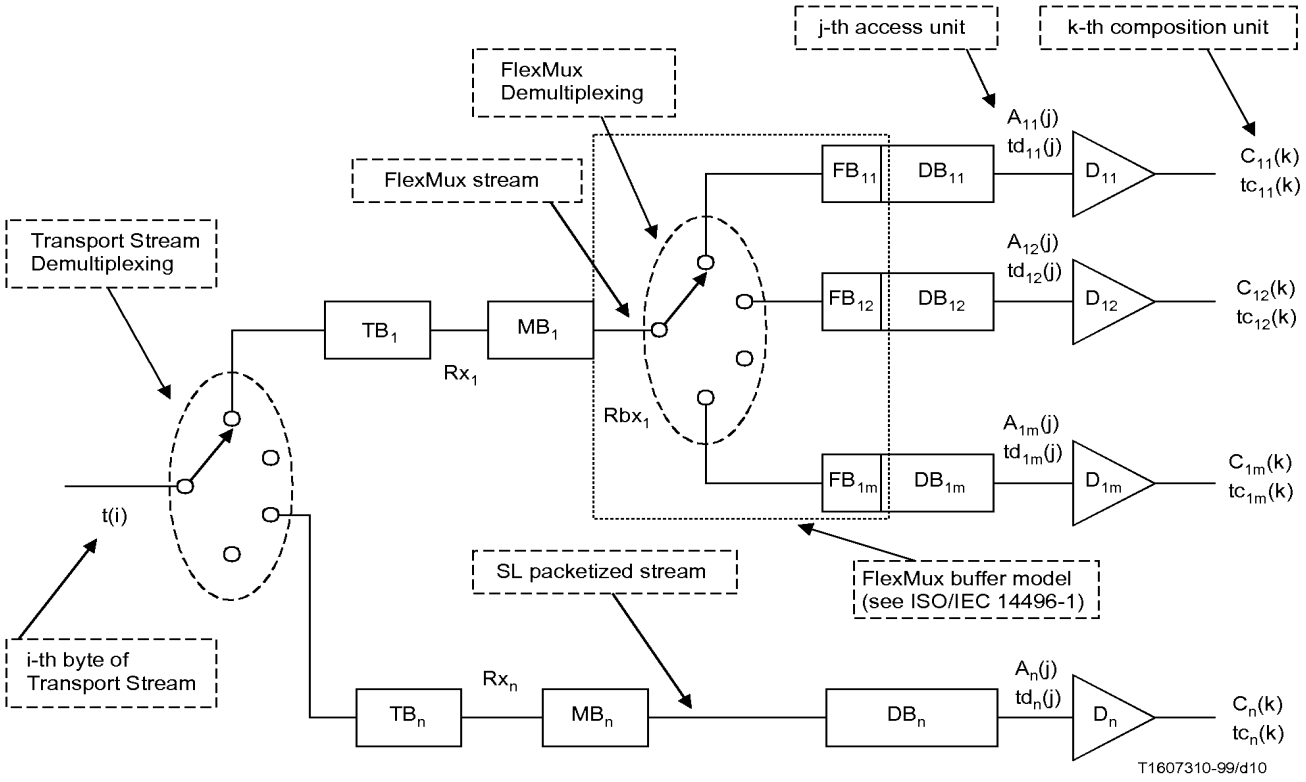


Figure 2-5 – T-STD model for ISO/IEC 14496 content

The following notation is used in Figure 2-5 and its description:

TB_n is the transport buffer.

MB_n is the multiplex buffer for FlexMux stream n or for SL-packetized stream n.

FB_{np} is the FlexMux buffer for the elementary stream in FlexMux channel p of FlexMux stream n.

DB_{np} is the decoder buffer for the elementary stream in FlexMux channel p of FlexMux stream n.

DB_n is the decoder buffer for elementary stream n.

D_{np} is the decoder for the elementary stream in FlexMux channel p of FlexMux stream n.

D_n is the decoder for elementary stream n.

Rx_n is the rate at which data are removed from TB_n.

Rbx_n is the rate at which data are removed from MB_n.

$A_{np}(j)$ is the jth access unit in elementary stream in FlexMux channel p of FlexMux stream n. $A_{np}(j)$ is indexed in decoding order.

$A_n(j)$ is the jth access unit in elementary stream n. $A_n(j)$ is indexed in decoding order.

$Td_{np}(j)$ is the decoding time, measured in seconds, in the system target decoder of the jth access unit in elementary stream in FlexMux channel p of FlexMux stream n.

$Td_n(j)$ is the decoding time, measured in seconds, in the system target decoder of the jth access unit in elementary stream n.

$C_{np}(k)$ is the kth composition unit in elementary stream in FlexMux channel p of FlexMux stream n. $C_{np}(k)$ results from decoding $A_{np}(j)$. $C_{np}(k)$ is indexed in composition order.

$C_n(k)$ is the kth composition unit in elementary stream n. $C_n(k)$ results from decoding $A_n(j)$. $C_n(k)$ is indexed in composition order.

$tc_{np}(k)$ is the composition time, measured in seconds, in the system target decoder of the kth composition unit in elementary stream in FlexMux channel p of FlexMux stream n.

$tc_n(k)$ is the composition time, measured in seconds, in the system target decoder of the kth composition unit in elementary stream n.

t(i) indicates the time in seconds at which the ith byte of the Transport Stream enters the system target decoder.

2.11.3.9.2 Processing of FlexMux streams

Complete Transport Stream packets containing data from FlexMux stream n are passed to the transport buffer for FlexMux stream n, TB_n. The size of TB_n is fixed at 512 bytes. All bytes that enter TB_n are removed from TB_n at a rate Rx_n, specified by the TB_leak_rate field in the MultiplexBuffer descriptor associated with FlexMux stream n. When there is no data in buffer TB_n, rate Rx_n is equal to zero. Duplicate Transport Stream packets are not delivered to MB_n.

In case of carriage in PES packets, the PES packet header and payload data bytes are delivered to buffer MB_n; all other bytes leaving TB_n do not enter MB_n, and may be used to control the system. In case of carriage in ISO_IEC_14496_sections, the section header, payload and CRC32 data bytes are delivered to buffer MB_n; all other bytes do not enter MB_n and may be used to control the system. In either case, the size of MB_n shall be specified by the MB_buffer_size field in the MultiplexBuffer descriptor.

The FlexMux Stream packet bytes in buffer MB_n are all delivered to their associated FlexMux buffer at the rate specified by the field fmxRate encoded in the FlexMux stream and in compliance with the FlexMux buffer model defined in subclause 11.2.9 of ISO/IEC 14496-1. Only FlexMux packet payload data bytes in FlexMux channel p of FlexMux stream n enter buffer FB_{np}. FlexMux packet header bytes in FlexMux channel p of FlexMux stream n are discarded and may be used to control the system. The rate specified by the fmxRate field shall be applicable for all FlexMux packets in the stream immediately following the FlexMux Clock Reference channel packet up to the next encountered FlexMux Clock Reference channel packet. When there is no FlexMux stream data present in MB_n, no data is removed from MB_n. Bytes from the PES packet header or from the ISO_IEC_14496_section header that immediately precede a FlexMux header are instantaneously removed and discarded and may be used to control the system. Bytes from the ISO_IEC_14496_section CRC32 fields that immediately follow the last FlexMux Stream packet in the section payload are removed instantaneously and discarded and may be used to verify the integrity of the data. Bytes from the FlexMux Clock Reference channel are instantaneously removed and discarded and may be used to lock the ISO/IEC 14496 object time base to the STC. When there is no PES packet or section payload data bytes, respectively present in MB_n, no data is removed from MB_n. All data that enters MB_n leaves it. All PES packet payload bytes of stream n enter the FlexMux demultiplexer instantaneously upon leaving MB_n.

2.11.3.9.3 Definition of FlexMux Buffer, FB_{np}

For each channel p of a FlexMux stream n, the size of FlexMux buffer FB_{np} is defined using the FmxBufferSize descriptor. FlexMux packet payload bytes are transferred from buffer FB_{np} to decoder buffer DB_{np} in compliance with the FlexMux buffer model defined in subclause 11.2.9 of ISO/IEC 14496-1. Only SL packet payload bytes in FlexMux channel p of FlexMux stream n enter buffer DB_{np}. The SL packet header bytes in FlexMux channel p of FlexMux stream n are discarded and may be used to control the system.

2.11.3.9.4 Processing of SL-packetized streams

Complete Transport Stream packets containing data from SL-packetized stream n are passed to the transport buffer for SL-packetized stream n, TB_n. All bytes that enter TB_n are removed at a rate Rx_n, specified by the TB_leak_rate field in the MultiplexBuffer descriptor. When there is no data in buffer TB_n, rate Rx_n is equal to zero. Duplicate Transport Stream packets are not delivered to MB_n.

In case of carriage in PES packets, the PES packet header and payload data bytes are delivered to buffer MB_n; all other bytes leaving TB_n do not enter MB_n, and may be used to control the system. In case of carriage in ISO_IEC_14496_sections, the section header, payload and CRC32 data bytes are delivered to buffer MB_n; all other bytes do not enter MB_n and may be used to control the system. In either case the size of MB_n is specified by the MB_buffer_size field in the MultiplexBuffer descriptor.

The SL-packetized stream bytes in buffer MB_n are all delivered to the decoder buffer DB_n at the rate specified by the field instantBitRate encoded in the SL-packetized stream and in compliance with the System Decoder Model defined in subclause 7.4 of ISO/IEC 14496-1. The rate specified by the instantBitRate field shall be applicable for all data bytes in the SL-packetized stream immediately following the instantBitRate field in the SL packet header up to the next encountered instantBitRate field. If there are no SL-packetized stream bytes in MB_n, no bytes are removed from MB_n. Bytes from the PES packet header or from the ISO_IEC_14496_section header that immediately precede a SL packet header are instantaneously removed and discarded and may be used to control the system. Bytes from the ISO_IEC_14496_section CRC32 fields that immediately follow the last SL packet payload byte in the section are removed instantaneously and discarded and may be used to verify the integrity of the data. When there are no PES packet or section payload data bytes, respectively present in MB_n, no data is removed from MB_n. All data that enters MB_n leaves it. All PES packet payload bytes of stream n enter buffer DB_n instantaneously upon leaving MB_n, with the exception of the SL packet headers. Bytes from the SL packet headers do not enter DB_n and may be used to control the system. The size of decoder buffer DB_n is given by the bufferSizeDB of the DecoderConfigDescriptor defined in ISO/IEC 14496-1.

2.11.3.9.5 Buffer management

Transport streams shall be constructed so that conditions defined in this subclause are satisfied.

TB_n shall not overflow and shall be empty at least once every second. MB_n shall not overflow. FB_{np} shall not overflow. DB_{np} and DB_n shall neither underflow nor overflow. Underflow of DB_{np} occurs when one or more bytes of an access unit are not present in DB_{np} at the decoding time associated with this access unit. Underflow of DB_n occurs when one or more bytes of an access unit are not present in DB_n at the decoding time associated with this access unit.

2.11.3.10 Carriage within a Transport Stream

2.11.3.10.1 Overview

A Transport Stream may contain one or more programs, each described by a Program Map Table. ISO/IEC 14496 content can be conveyed in addition to the already defined stream types for such a program. Elements of the ISO/IEC 14496 content may be conveyed in one or more ITU-T Rec. H.222.0 | ISO/IEC 13818-1 program elements referenced by a unique PID value within a Transport Stream. As a special case, it is possible that a program within a Transport Stream consists only of ISO/IEC 14496 program elements. ISO/IEC 14496 content associated to a program and carried in the Transport Stream shall be referenced in the Program Map Table of that program. An initial object descriptor shall be used to define an ISO/IEC 14496-1 scene; the use of this descriptor is specified in subclause 2.11.3.10.2.

Carriage of ISO/IEC 14496 content in a PID is signaled by a stream_type value of 0x12 or 0x13 in the Program Map Table in association with that PID value. A value of 0x12 indicates carriage in PES packets. The stream_id field in the PES packet header signals whether the PES packet contains a single SL packet or a number of FlexMux packets. A stream_type value of 0x13 in the Program Map Table indicates that the program element carries an object descriptor

stream or a BIFS-Command stream contained in sections. In this case the table_id in the section header indicates whether an object descriptor stream is carried in the sections or a BIFS-Command stream. See also table 2-65. The section contains either a single SL packet or a number of FlexMux packets, as indicated by the presence of an SL descriptor or a FMC descriptor respectively in the descriptor loop of the Program Map Table for the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 program element that carries the sections. When ISO/IEC 14496 content is carried, the SL descriptor and the FMC descriptor shall specify the ES_ID for each encapsulated ISO/IEC 14496 stream. When the assignment of ES_ID values changes, the Program Map Table shall be updated and the version_number of the PMT shall be incremented by 1 modulo 32. An example of a content access procedure for ISO/IEC 14496 program components within a Transport Stream is given in Annex R Carriage of ISO/IEC 14496 scenes in ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

Table 2-65 – ISO/IEC defined options for carriage of an ISO/IEC 14496 scene and associated streams in ITU-T Rec. H.222.0 | ISO/IEC 13818-1

ISO/IEC 14496-1 object descriptor streams	Encapsulation in SL packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1010'
		Carriage in ISO_IEC_ 14496_sections	Stream_type = 0x13	Table_id = 0x05
	Encapsulation in SL packets followed by Multiplex into FlexMux packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1011'
		Carriage in ISO_IEC_ 14496_sections	Stream_type = 0x13	Table_id = 0x05
ISO/IEC 14496-1 scene description streams	Encapsulation in SL packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1010'
		Carriage in ISO_IEC_ 14496_sections	Stream_type = 0x13	Table_id = 0x04
	Encapsulation in SL packets followed by Multiplex into FlexMux packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1011'
		Carriage in ISO_IEC_ 14496_sections	Stream_type = 0x13	Table_id = 0x04
All other ISO/IEC 14496 streams	Encapsulation in SL packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1010'
	Encapsulation in SL packets followed by Multiplex into FlexMux packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1011'

2.11.3.10.2 Initial Object Descriptor

In case of carriage of an ISO/IEC 14496-1 scene, the ISO/IEC 14496-1 initial object descriptor serves as the initial access point to all associated streams. The initial object descriptor shall be conveyed in the IOD descriptor located in the descriptor loop immediately following the program_info_length field in the Program Map Table of the program to which the scene is associated. It contains ES_Descriptors identifying the scene description and object descriptor streams that form part of this program. It may also contain ES_Descriptors identifying one or more associated IPMP or OCI streams. Identification of streams is done by means of ES_IDs as specified in clause 8 of ISO/IEC 14496-1.

2.11.3.11 P-STD Model for 14496 content

Figure 2-6 shows the STD model when ISO/IEC 14496 systems data are carried in a Program Stream.

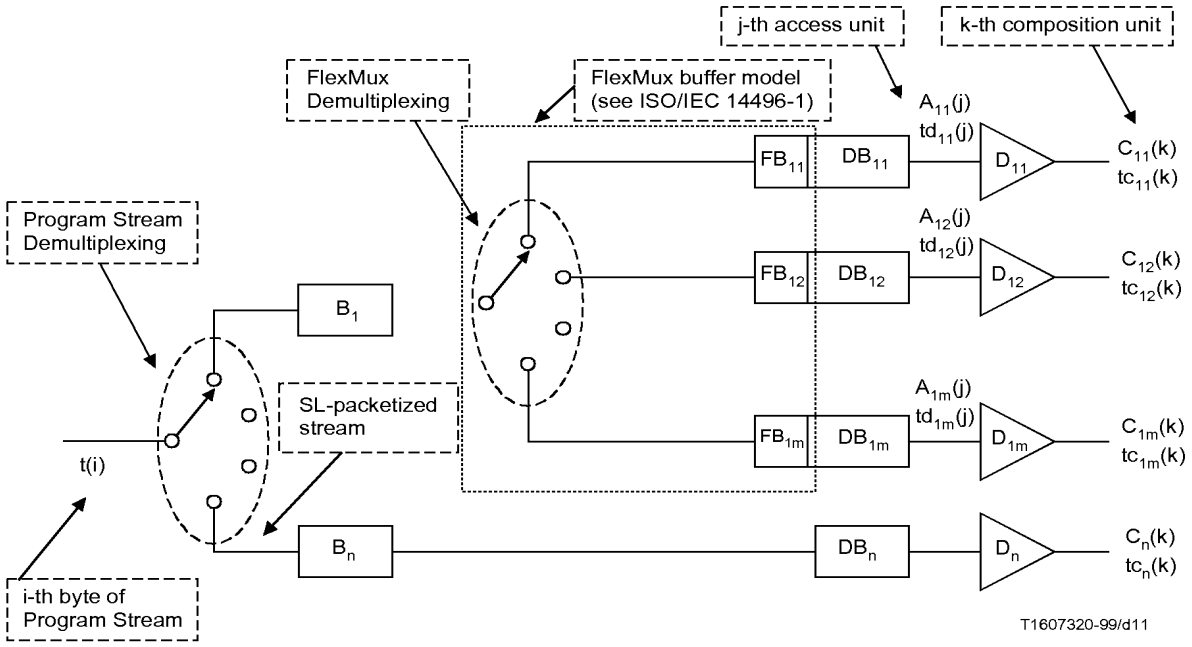


Figure 2-6 – P-STD model for ISO/IEC 14496 Systems stream

The following notation is used in Figure 2-6 and its description :

B_n is the input buffer for FlexMux stream n or for SL-packetized stream n.

FB_{np} is the FlexMux buffer for the elementary stream in FlexMux channel p of FlexMux stream n.

DB_{np} is the decoder buffer for the elementary stream in FlexMux channel p of FlexMux stream n.

DB_n is the decoder buffer for elementary stream n.

D_{np} is the decoder for elementary stream in FlexMux channel p of FlexMux stream n.

D_n is the decoder for elementary stream n.

$A_{np}(j)$ is the jth access unit in elementary stream in FlexMux channel p of FlexMux stream n. $A_{np}(j)$ is indexed in decoding order.

$A_n(j)$ is the jth access unit in elementary stream n. An(j) is indexed in decoding order.

$Td_{np}(j)$ is the decoding time, measured in seconds, in the system target decoder of the jth access unit in elementary stream in FlexMux channel p of FlexMux stream n.

$Td_n(j)$ is the decoding time, measured in seconds, in the system target decoder of the jth access unit in elementary stream n.

$C_{np}(k)$ is the kth composition unit in elementary stream in FlexMux channel p of FlexMux stream n. $C_{np}(k)$ results from decoding $A_{np}(j)$. $C_{np}(k)$ is indexed in composition order.

$C_n(k)$ is the kth composition unit in elementary stream n. $C_n(k)$ results from decoding $A_n(j)$. $C_n(k)$ is indexed in composition order.

$tc_{np}(k)$ is the composition time, measured in seconds, in the system target decoder of the kth composition unit in elementary stream in FlexMux channel p of FlexMux stream n.

$tc_n(k)$ is the composition time, measured in seconds, in the system target decoder of the kth composition unit in elementary stream n.

t(i) indicates the time in seconds at which the ith byte of the Program Stream enters the system target decoder.

2.11.3.11.1 Processing of FlexMux streams

At the input of the STD each byte in the payload of PES packets carrying a FlexMux stream n is transferred instantaneously to buffer B_n. The i-th byte enters B_n at time t(i). PES packet header bytes do not enter buffer B_n and may be used to control the system. The size of B_n is specified by the P-STD_buffer_size field in the header of the PES packet that carries stream n.

The FlexMux stream packet bytes in buffer B_n are all delivered to their associated FlexMux buffer at the rate specified by the field fmxRate encoded in the FlexMux stream and in compliance with the FlexMux buffer model defined in clause 11.2.9 of ISO/IEC 14496-1. Only FlexMux packet payload data bytes in FlexMux channel p of FlexMux stream n enter buffer FB_{np}. FlexMux packet header bytes in FlexMux channel p of FlexMux stream n are discarded and may be used to control the system. The rate specified by the fmxRate field shall be applicable for all FlexMux packets in the stream up to the next encountered FlexMux Clock Reference channel packet. Bytes from the FlexMux Clock Reference channel are instantaneously removed and discarded and may be used to lock the ISO/IEC 14496 object time base to the STC. When there is no PES packet payload data present in B_n, no data is removed from B_n. All data that enters B_n leaves it. All PES packet payload bytes of stream n enter the FlexMux demultiplexer instantaneously upon leaving B_n.

2.11.3.11.2 Definition of FlexMux Buffer, FB_{np}

For each channel p of a FlexMux stream n, the size of FlexMux buffer FB_{np} is defined using the FmxBufferSize descriptor if a Program Stream Map is present in the Program Stream. FlexMux packet payload bytes are transferred from buffer FB_{np} to decoder buffer DB_{np} in compliance with the FlexMux buffer model defined in subclause 11.2.9 of ISO/IEC 14496-1. Only SL packet payload bytes in FlexMux channel p of FlexMux stream n enter buffer DB_{np}. The SL packet header bytes in FlexMux channel p of FlexMux stream n are discarded and may be used to control the system

2.11.3.11.3 Processing of SL-packetized streams

At the input of the STD each byte in the payload of PES packets carrying an SL-packetized stream n is transferred instantaneously to buffer B_n. The i-th byte enters B_n at time t(i). PES packet header bytes do not enter buffer B_n and may be used to control the system. The size of B_n is specified by the P-STD_buffer_size field in the header of the PES packet that carries stream n. The SL-packetized stream bytes in buffer B_n are delivered to the decoder buffer DB_n at the rate specified by the field instantBitRate encoded in the SL-packetized stream and in compliance with the System Decoder Model defined in subclause 7.4 of ISO/IEC 14496-1. The rate specified by the instantBitRate field shall be applicable for all data bytes in the SL-packetized stream up to the next encountered instantBitRate field. When there is no PES packet payload data present in B_n, no data is removed from B_n. All data that enters B_n leaves it. All bytes of stream n enter buffer DB_n instantaneously upon leaving B_n, with the exception of the SL packet headers. Bytes from the SL packet headers do not enter DB_n and may be used to control the system. The size of decoder buffer DB_n is given by the bufferSizeDB of the DecoderConfigDescriptor defined in ISO/IEC 14496-1.

2.11.3.11.4 Buffer management

Program Streams shall be constructed so that B_n does not overflow. FB_{np} shall not overflow. DB_{np} and DB_n shall neither underflow nor overflow. Underflow of DB_{np} occurs when one or more bytes of an access unit are not present in DB_{np} at the decoding time associated with this access unit. Underflow of DB_n occurs when one or more bytes of an access unit are not present in DB_n at the decoding time associated with this access unit.

2.11.3.12 Carriage within a Program Stream

2.11.3.12.1 Overview

A Program Stream contains only one program. ISO/IEC 14496 data can be conveyed in addition to the already defined stream types for such a program. As a special case, it is also possible that a Program Stream carries only ISO/IEC 14496 data. If a Program Stream Map is present, ISO/IEC 14496 content carried in the Program Stream shall be referenced as follows. Carriage of ISO/IEC 14496-1 scenes and associated ISO/IEC 14496 streams in SL and FlexMux packets is indicated by the appropriate stream_id and by an initial object descriptor; the use of this descriptor is specified in subclause 2.11.3.12.2. For each carried ISO/IEC 14496 stream the SL descriptor and the FMC descriptor shall specify the ES_ID. When the assignment of ES_ID values changes, the Program Stream Map, if present, shall be updated and the program_stream_map_version shall be incremented by 1 modulo 32. Note that in a Program Stream the ISO/IEC 14496 content may also be referenced by private means.

For an example of a content access procedure for ISO/IEC 14496 program components within a Program Stream, see Annex R.

2.11.3.12.2 Initial Object Descriptor

In case of carriage of an ISO/IEC 14496-1 scene, the ISO/IEC 14496 initial object descriptor serves as the initial access point to all associated streams. If a Program Stream Map is present in the Program Stream, the initial object descriptor shall be conveyed in the IOD descriptor that is located in the descriptor loop immediately following the program_stream_info_length field. It contains ES_Descriptors identifying the scene description and object descriptor streams of the scene that form part of this program. It may also contain ES_Descriptors identifying one or more associated IPMP or OCI streams. Identification of streams is done by means of ES_IDs as specified in clause 8 of ISO/IEC 14496-1. In a Program Stream, the initial object descriptor may also be conveyed by private means.

Annex A

CRC Decoder Model

(This annex forms an integral part of this Recommendation | International Standard)

A.0 CRC decoder model

The 32-bit CRC Decoder Model is specified in Figure A.1.

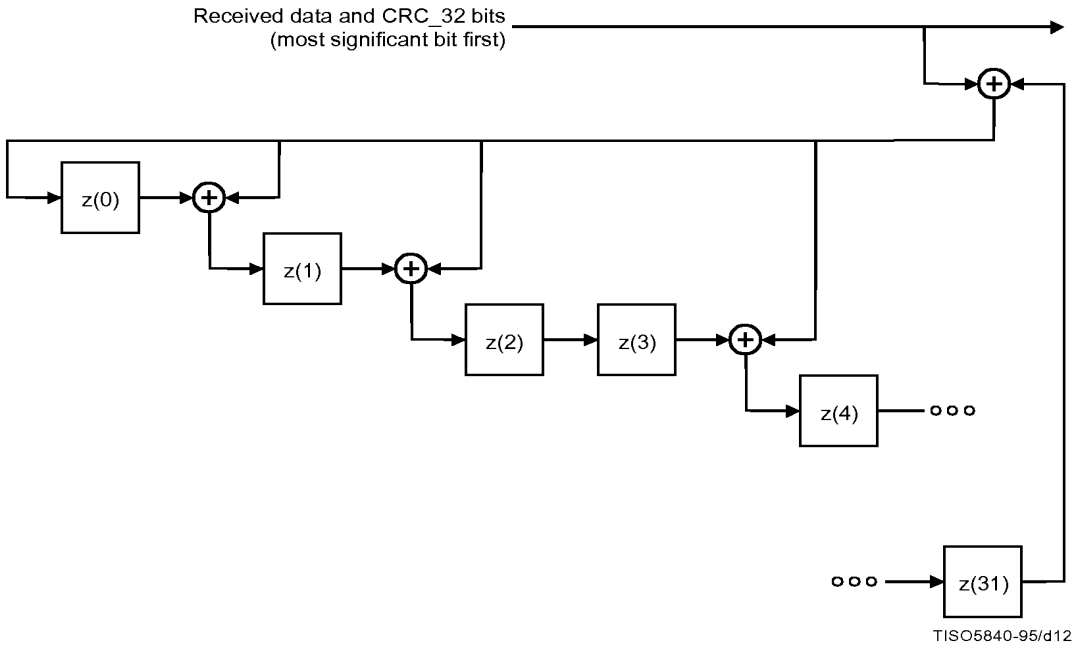


Figure A.1 – 32-bit CRC decoder model

The 32-bit CRC Decoder operates at bit level and consists of 14 adders '+' and 32 delay elements z(i). The input of the CRC decoder is added to the output of z(31), and the result is provided to the input z(0) and to one of the inputs of each remaining adder. The other input of each remaining adder is the output of z(i), while the output of each remaining adder is connected to the input of z(i + 1), with i = 0, 1, 3, 4, 6, 7 , 9, 10, 11, 15, 21, 22, and 25. Refer to Figure A.1 above.

This is the CRC calculated with the polynomial:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \qquad \text{(A-1)}$$

Bytes are received at the input of the CRC decoder. Each byte is shifted into the CRC decoder one bit at a time, with the left most bit (msb) first. For example, if the input is byte 0x01 the seven '0's enter the CRC decoder first, followed by the one '1'. Before the CRC processing of the data of a section the output of each delay element z(i) is set to its initial value '1'. After this initialization, each byte of the section is provided to the input of the CRC decoder, including the four CRC_32 bytes. After shifting the last bit of the last CRC_32 byte into the decoder, i.e. into z(0) after the addition with the output of z(31), the output of all delay elements z(i) is read. In the case where there are no errors, each of the outputs of z(i) shall be zero. At the CRC encoder the CRC_32 field is encoded with a value such that this is ensured.

Annex B

Digital Storage Medium Command and Control (DSM-CC)

(This annex does not form an integral part of this Recommendation | International Standard)

B.0 Introduction

The DSM CC protocol is a specific application protocol intended to provide the basic control functions and operations specific to managing an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream on digital storage media. This DSM CC is a low-level protocol above network/OS layers and below application layers.

The DSM-CC shall be transparent in the following sense:

- it is independent of the DSM used;
- it is independent of whether the DSM is located at a local or remote site;
- it is independent of the network protocol with which the DSM-CC is interfaced;
- it is independent of the various operating systems on which the DSM is operated.

B.0.1 Purpose

Many applications of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 DSM Control Commands require access to an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream stored on a variety of digital storage media at a local or remote site. Different DSM have their own specific control commands and thus, a user would need to know different sets of specific DSM control commands in order to access ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstreams from different DSM. This brings many difficulties to the interface design of an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 or ISO 11172-1 application system. To overcome this difficulty, a set of common DSM control commands, which is independent of the specific DSM used, is suggested in this annex. This annex is informative only. ISO/IEC 13818-6 defines DSM-CC extension with a broader scope.

B.0.2 Future applications

Beyond the immediate applications supported by the current DSM control commands, future applications based on extensions of DSM command control could include the following:

Video on demand

Video programs are provided as requested by a customer through various communication channels. The customer could select a video program from a list of programs available from a video server. Such applications could be used by hotels, cable TV, educational institutions, hospitals, etc.

Interactive video services

In these applications, the user provides frequent feedback controlling the manipulation of stored video and audio. These services can include video based games, user controlled video tours, electronic shopping, etc.

Video networks

Various applications may wish to exchange stored audio and video data through some type of computer network. Users could route AV information through the video network to their terminals. Electronic publishing and multimedia applications are examples of this kind of application.

B.0.3 Benefits

Specifying the DSM control commands independent of the DSM, end-users can perform ITU-T Rec. H.222.0 | ISO/IEC 13818-1 decoding without having to fully understand the detailed operation of the specific DSM used.

The DSM control commands are codes to give end users the assurance that the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstreams can be played and stored with the same semantics, independent of the DSM and user interface. They are fundamental commands for the control of DSM operation.

B.0.4 Basic functions

B.0.4.1 Stream selection

The DSM-CC provides the means to select an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream upon which to perform the succeeding operations. Such operations include creation of a new bitstream. Parameters of this function include:

- index of the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream (the mapping between this index and a name meaningful to an application is outside the scope of the current DSM-CC)
- mode (retrieval/storage)

B.0.4.2 Retrieval

The DSM CC provides the means to:

- play an identified ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream;
- play from a given presentation time;
- set the playback speed (normal or fast);
- set the playback duration (until a specified presentation time, the end of the bitstream in forward play or the beginning in reverse play or the issuance of a stop command);
- set the direction (forward or reverse);
- pause;
- resume;
- change the access point in the bitstream;
- stop.

B.0.4.3 Storage

The DSM-CC provides the means to:

- cause storage of a valid bitstream for a specified duration;
- cause storage to stop.

DSM-CC provides a useful but limited subset of functionality that may be required in DSM based ITU-T Rec. H.222.0 | ISO/IEC 13818-1 applications. It is fully expected that significant additional capabilities will be added through subsequent extensions.

B.1 General elements

B.1.1 Scope

The scope of this work consists of the development of a Recommendation | International Standard to specify a useful set of commands for control of digital storage media on which an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream is stored. The commands can perform remote control of a digital storage media in a general way independently of the specific DSM and apply to any ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream stored on a DSM.

B.1.2 Overview of the DSM-CC application

The current DSM-CC syntax and semantics cover the single user to DSM application. The user's system is capable of retrieving an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream and is also (optionally) capable of generating an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream. The control channel over which the DSM commands and acknowledgements are sent is shown in Figure B.1 as an out of band channel. This can also be accomplished by inserting the DSM-CC commands and acknowledgements into the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstreams if an out of band channel is not available.

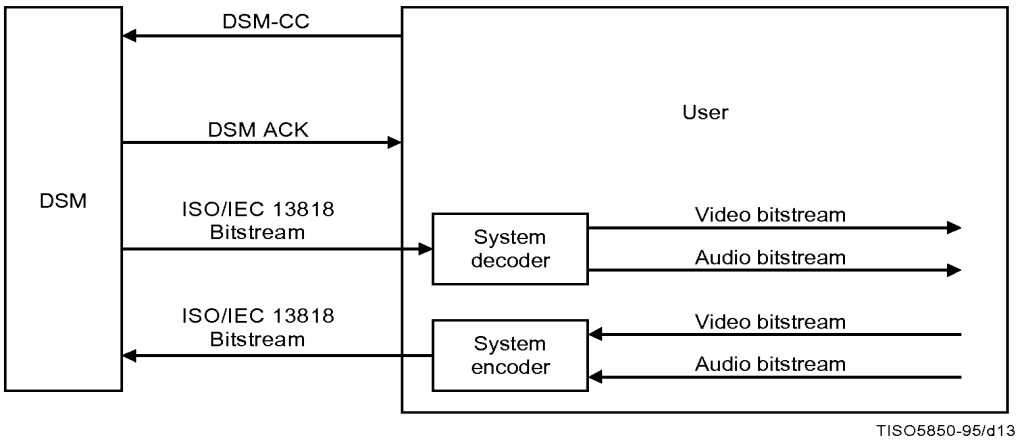


Figure B.1 – Configuration of DSM-CC application

B.1.3 The transmission of DSM-CC commands and acknowledgements

The DSM-CC is encoded into a DSM-CC bitstream according to the syntax and semantics defined in B.2.2 through B.2.9. The DSM-CC bitstream can be transmitted both as a stand-alone bitstream and in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Systems bitstream.

When the DSM-CC bitstream is transmitted in stand alone mode, its relationship to the Systems bitstream and the decoding process is illustrated in Figure B.2. In this case, the DSM-CC bitstream is not embedded in the Systems bitstream. This transmission mode can be used in the applications when the DSM is connected directly with the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 decoder. It can also be used in the applications where the DSM-CC bitstream could be controlled and transmitted by other types of network multiplexors.

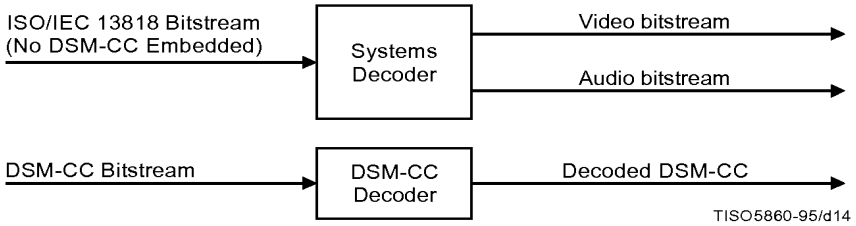


Figure B.2 – DSM-CC bitstream decoded as a standalone bitstream

For some applications, it is desirable to transmit the DSM-CC in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 systems bitstream so that some features of the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 systems bitstream could be applied to the DSM-CC bitstream as well. In this case, the DSM-CC bitstream is embedded in the systems bitstream by the systems multiplexor.

The DSM-CC bitstream is encoded by the systems encoder in the following process. First, the DSM-CC bitstream is packetized into a packetized elementary stream (PES) according to the syntax described in 2.4.3.6. The PES packet is then multiplexed into either a Program Stream (PS) or a Transport Stream (TS) according to the requirement of the transmission media. The decoding procedures are the inverse of the encoding procedures and are illustrated in the block diagram of the Systems decoder depicted in Figure B.3.

In Figure B.3, the output of the Systems decoder is a video bitstream, audio bitstream and/or DSM-CC bitstream. The DSM-CC bitstream is identified by the stream_id, value '1111 0010' as defined by the stream_id Table 2-18. Once the DSM-CC bitstream is identified, it follows the rules as specified by T-STD or P-STD.

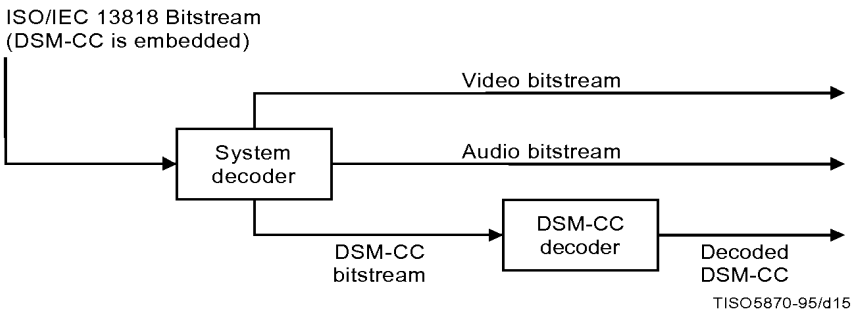


Figure B.3 – DSM-CC bitstream decoded as part of the system bitstream

B.2 Technical elements

B.2.1 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply:

B.2.1.1 DSM-CC: Digital Storage Media Command and Control Commands that are specified by Rec. H.222.0 | ISO/IEC 13818-1 for the control of digital storage media at a local or remote site containing an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream.

B.2.1.2 DSM ACK: The acknowledgement from the DSM-CC command receiver to the command initiator.

B.2.1.3 MPEG bitstream: An ISO/IEC 11172-1 Systems stream, ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Stream or ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Transport stream.

B.2.1.4 DSM-CC server: A system, either local or remote, used to store and/or retrieve an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream.

B.2.1.5 point of random access: A point in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream with the property that for at least one elementary stream within the bitstream, the next access unit, 'N', completely contained in the bitstream can be decoded without reference to previous access units, and for every elementary stream in the bitstream all access units with the same or later presentation times are completely contained subsequently in the bitstream and can be completely decoded by a system target decoder without access to information prior to the point of random access. The bitstream as stored on the DSM may have certain points of random access; the output of the DSM may include additional points of random access manufactured by the DSM's own manipulation of the stored material (e.g. storing quantization matrices so that a sequence header can be generated whenever necessary). A point of random access has an associated PTS, namely the actual or implied PTS of access unit 'N'.

B.2.1.6 current operational PTS value: The actual or implied PTS associated with the last point of random access preceding the last access unit provided from the DSM from the currently selected ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream. If no access unit has been provided from this ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream, the DSM is incapable of providing random access into the current bitstream, then the current operational PTS value is the first point of random access in the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream.

B.2.1.7 DSM-CC bitstream: A sequence of bits satisfying the syntax of B.2.2.

B.2.2 Specification of DSM-CC syntax

- Every DSM control command shall commence with a start_code, as specified in Table B.1.
- Every DSM control command shall have a packet_length to specify the number of byte in a DSM-CC packet.
- When the DSM-CC bitstream is transmitted as a PES packet as defined in 2.4.3.6, the fields up to the packet_length field are identical to those specified in 2.4.3.6. In other words, if the DSM-CC packet is encapsulated in a PES packet, the PES packet start code is the only start code at the beginning of the packet.

- The actual control command or acknowledgement shall follow the last byte of the packet_length field.
- An acknowledgement stream shall be provided by the DSM control bitstream receiver after the requested operation is started or is completed, depending on the command received.
- At all times the DSM is responsible for providing a normative ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream. This may include manipulating the trick mode bits defined in 2.4.3.6.

Table B.1 – DSM-CC syntax

Syntax	No. of bits	Mnemonic
DSM_CC() {		
packet_start_code_prefix	**24**	**bslbf**
stream_id	**8**	**uimsbf**
packet_length	**16**	**uimsbf**
command_id	**8**	**uimsbf**
If (command_id == '01') {		
control()		
} else if (command_id == '02') {		
ack()		
}		
}		

B.2.3 Semantics of fields in specification of DSM-CC syntax

packet_start_code_prefix – This is a 24-bit code. Together with the stream_id that follows it constitutes a DSM-CC packet start code that identifies the beginning of a DSM-CC packet bitstream. The packet_start_code_prefix is the bit string '0000 0000 0000 0000 0000 0001' (0x000001).

stream_id – This 8-bit field specifies the bitstream type and shall have a value '1111 0010' for the DSM-CC bitstream. Refer to Table 2-19.

packet_length – This 16-bit field specifies the number of bytes in the DSM-CC packet immediately following the last byte of this field.

command_id – This 8-bit unsigned integer identifies the bitstream is a control command or an acknowledgement stream. The values are defined in Table B.2.

Table B.2 – Command_id assigned values

Value	**Command_id**
0x00	Forbidden
0x01	Control
0x02	Ack
0x03-0xFF	Reserved

B.2.4 Control layer

Constraints on setting flags in DSM-CC control

- At most one of the flags for select, playback and storage shall be set to '1' for each DSM control command. If none of these bits are set, then this command shall be ignored.
- At most one of pause_mode, resume_mode, stop_mode, play_flag, and jump_flag shall be set for each retrieval command. If none of these bits are set, then this command shall be ignored.
- At most one of record_flag and stop_mode shall be selected for each storage command. If none of these bits are set, then this command shall be ignored.

See Table B.3.

Table B.3 – DSM-CC control

Syntax	No. of bits	Mnemonic
control() {		
select_flag	**1**	**bslbf**
retrieval_flag	**1**	**bslbf**
storage_flag	**1**	**bslbf**
reserved	**12**	**bslbf**
marker_bit	**1**	**bslbf**
If (select_flag == '1') {		
bitstream_id [31..17]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
bitstream_id [16..2]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
bitstream_id [1..0]	**2**	**bslbf**
select_mode	**5**	**bslbf**
marker_bit	**1**	**bslbf**
}		
if (retrieve_flag == '1') {		
jump_flag	**1**	**bslbf**
play_flag	**1**	**bslbf**
pause_mode	**1**	**bslbf**
resume_mode	**1**	**bslbf**
stop_mode	**1**	**bslbf**
reserved	**10**	**bslbf**
marker_bit	**1**	**bslbf**
if (jump_flag == '1') {		
reserved	**7**	**bslbf**
direction_indicator	**1**	**bslbf**
time_code()		
}		
if (play_flag == '1'){		
speed_mode	**1**	**bslbf**
direction_indicator	**1**	**bslbf**
reserved	**6**	**bslbf**
time_code()		
}		
}		
if (storage_flag == '1') {		
reserved	**6**	**bslbf**
record_flag	**1**	**bslbf**
stop_mode	**1**	**bslbf**
if (record_flag == '1') {		
time_code()		
}		
}		
}		

B.2.5 Semantics of fields in control layer

marker_bit – This is a 1-bit marker that is always set to '1' to avoid start code emulation.

reserved_bits – This 12-bit field is reserved for future use by this Recommendation | International Standard for DSM control commands. Until otherwise specified by ITU-T | ISO/IEC it shall have the value '0000 0000 0000'.

select_flag – This 1-bit flag when set to '1' specifies a bitstream selection operation. When it is set to '0' no bitstream selection operation shall occur.

retrieval_flag – This 1-bit flag when set to '1' specifies that a specific retrieval (playback) action will occur. The operation starts from the current operational PTS value.

storage_flag – This 1-bit flag when set to '1' specifies that a storage operation is to be executed.

bitstream_ID – This 32-bit field is coded in three parts. The parts are combined to form an unsigned integer specifying which ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream is to be selected. It is the DSM server's responsibility to map the names of the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstreams stored on its DSM uniquely to a series of numbers which could be represented by the bitstream_ID.

select_mode – This 5-bit unsigned integer specifies which mode of bitstream operation is requested. Table B.4 specifies the defined modes.

Table B.4 – Select mode assigned values

Code	Mode
0x00	Forbidden
0x01	Storage
0x02	Retrieval
0x03-0x1F	Reserved

jump_flag – This 1-bit flag when set to '1' specifies a jump in the playback pointer to a new access unit. The new PTS is specified by a relative time_code with respect to the current operational PTS value. This function is only valid when the current ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream is in the "stop" mode.

play_flag – This 1-bit flag when set to '1' specifies to play a bitstream for a certain time period. The speed, direction, and play duration are additional parameters in the bit stream. The play starts from the current operational PTS value.

pause_mode – This is a one-bit code specifying to pause the playback action and keep the playback pointer at the current operational PTS value.

resume_mode – This is a one-bit code specifying to continue the playback action from the current operational PTS value. Resume only has meaning if the current bitstream is in the "pause" state, and the bitstream will be set to the forward play state at normal speed.

stop_mode – This is a one-bit code specifying to stop a bitstream transmission.

direction_indicator – This is a one-bit code to indicate the playback direction. If this bit is set to '1', it stands for a forward play. Otherwise it stands for a backward play.

speed_mode – This is a 1-bit code to specify the speed scale. If this bit is set to '1', it specifies that the speed is normal play. If this bit is set to '0', it specifies that the speed is fast play (i.e. fast forward or fast reverse).

record_flag – This is one-bit flag to specify the request of recording the bitstream from an end user to a DSM for a specified duration or until the reception of a stop command, whichever comes first.

B.2.6 Acknowledgement layer

Constraints on setting flags in DSM-CC control

Only one of the acks bits specified below can be set to '1' for each DSM ack bitstream (see Table B.5).

Table B.5 – DSM-CC Acknowledgement

Syntax	No. of bits	Mnemonic
ack() {		
select_ack	**1**	**bslbf**
retrieval_ack	**1**	**bslbf**
storage_ack	**1**	**bslbf**
error_ack	**1**	**bslbf**
reserved	**10**	**bslbf**
marker_bit	**1**	**bslbf**
cmd_status	**1**	**bslbf**
If (cmd_status == '1' &&		
(retrieval_ack == '1' || storage_ack == '1')) {		
time_code()		
}		
}		

B.2.7 Semantics of fields in Acknowledgement layer

select_ack – This 1-bit field when it is set to '1' indicates that the ack() command is to acknowledge a select command.

retrieval_ack – This 1-bit field when set to '1' indicates that the ack() command is to acknowledge a retrieval command.

storage_ack – This 1-bit field when set to '1' indicates that the ack() command is to acknowledge a storage command.

error_ack – This 1-bit field when set to '1' indicates a DSM error. The defined errors are EOF (end of file on forward play or start of file on reverse play) on a stream being retrieved and Disk Full on a stream being stored. If this bit is set to '1', cmd_status is undefined. The current bitstream is still selected.

cmd_status – This 1-bit flag set to '1' indicates that the command is accepted. When set to '0' it indicates the command is rejected. The semantics vary according to the command received as follows:

- If select_ack is set and cmd_status is set to '1', it specifies that the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream is selected and the server is ready to provide the selected mode of operation. The current operational PTS value is set to the first point of random access of the newly selected ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream. If cmd_status is set to '0', the operation has failed and no bitstream is selected.

- If retrieval_ack is set and cmd_status is set to '1', it specifies that the retrieval operation is initiated for all retrieval commands. The position of the current operational PTS pointer is reported by the succeeding time_code.

- For the play_flag command with infinite_time_flag != '1', a second acknowledgement will be sent. This will acknowledge that the play operation has ended by reaching the duration defined by the play_flag command.

- If the cmd_status is set to '0' in a retrieval acknowledgement, the operation has failed. Possible reasons for this failure include an invalid bitstream_ID, jumping beyond the end of a file, or a function not supported such as reverse play in standard speed.

- If storage_ack is set, it specifies that the storage operation is being started for the record_flag command or is completed by the stop_mode command. The PTS of the last complete access unit stored is reported by the succeeding time_code.

- If the recording operation is ended by reaching the duration defined by the storage_flag command, another acknowledgement shall be sent and the current operational PTS value after the recording shall be reported.

- If the cmd_status is set to '0' in a storage acknowledgement, the operation has failed. Possible reasons for this failure include an invalid bitstream_ID, or the inability of the DSM to store data.

B.2.8 Time code

Constraints on time code

- A forward operation of specified duration given by a time_code terminates after the actual or implied PTS of an access unit is observed such that PTS minus the current operational PTS value at the start of the operation modulo 2^{33} exceeds the duration.
- A backward operation of specified duration given by a time_code terminates after the actual or implied PTS of an access unit is observed such that current operational PTS value at the start of the operation minus that PTS modulo 2^{33} exceeds the duration.
- For all the commands in the control() layer, the time_code is specified as a relative duration with respect to the current operational PTS value.
- For all the commands in the ack() layer, the time_code is specified by the current operational PTS value.

See Table B.6.

Table B.6 – Time code

Syntax	No. of bits	Mnemonic
time_code() {		
reserved	**7**	**bslbf**
infinite_time_flag	**1**	**bslbf**
if (infinite_time_flag == '0') {		
reserved	**4**	**bslbf**
PTS [32..30]	**3**	**bslbf**
marker	**1**	**bslbf**
PTS [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
}		

B.2.9 Semantics of fields in time code

infinite_time_flag – This 1-bit flag when set to '1' indicates an infinite time period. This flag is set to '1' in applications where a time period for a specific operation could not be defined in advance.

PTS [32..0] – The presentation timestamp of the access unit of the bitstream. Depending upon the function, this can be an absolute value or a relative time delay in cycles of the 90 kHz system clock.

Annex C

Program Specific Information

(This annex does not form an integral part of this Recommendation | International Standard)

C.0 Explanation of Program Specific Information in Transport Streams

Subclause 2.4.4 contains the normative syntax, semantics and text concerning Program Specific Information. In all cases, compliance with the constraints of 2.4.4 is required. This annex provides explanatory information on how to use the PSI functions, and considers examples of how it may be used in practice.

C.1 Introduction

This Recommendation | International Standard provides a method for describing the contents of Transport Stream packets for the purpose of the demultiplexing and presentation of programs. The coding specification accommodates this function through the Program Specific Information (PSI). This annex discusses the use of PSI.

The PSI may be thought of as belonging to four tables:

1) Program Association Table (PAT);
2) TS Program Map Table (PMT);
3) Network Information Table (NIT);
4) Conditional Access Table (CAT).

The contents of the PAT, PMT and CAT are specified in this Recommendation | International Standard. The NIT is a private table, but the PID value of the Transport Stream packets which carry it is specified in the PAT. It must however, follow the section structure defined in this Recommendation | International Standard.

C.2 Functional Mechanism

The tables listed above are conceptual in that they need never be regenerated in a specified form within a decoder. While these structures may be thought of as simple tables, they may be partitioned before they are sent in Transport Stream packets. The syntax supports this operation by allowing the tables to be partitioned into sections and by providing a normative mapping method into Transport Stream packet payloads. A method is also provided to carry private data in a similar format. This is advantageous as the same basic processing in the decoder can then be used for both the PSI data and the private data helping to keep cost down. For advice on the optimum placing of PSI in the Transport Stream, see Annex D.

Each section is uniquely identified by the combination of the following elements:

i) **table_id**

The 8-bit table_id identifies to which table the section belongs.

- Sections with table_id 0x00 belong to the Program Association Table.
- Sections with table_id 0x01 belong to the Conditional Access Table.
- Sections with table_id 0x02 belong to the TS Program Map Table.

Other values of the table_id can be allocated by the user for private purposes.

It is possible to set up filters looking at the table_id field to identify whether a new section belongs to a table of interest or not.

ii) **table_id_extension**

This 16-bit field exists in the long version of a section. In the Program Association Table it is used to identify the transport_stream_id of the stream – effectively a user-defined label which allows one Transport Stream to be distinguished from another within a network or across networks. In the Conditional Access Table this field currently has no meaning and is therefore marked as "reserved" meaning that it shall be coded as 0xFFFF, but that a meaning may be defined by ITU-T | ISO/IEC in a subsequent revision of this Recommendation | International Standard. In a TS Program Map section the field contains the program_number, and thereby identifies the program to which the data in the section refers. The table_id_extension can also be used as a filter point in certain cases.

iii) **section_number**

The section_number field allows the sections of a particular table to be reassembled in their original order by the decoder. There is no obligation within this Recommendation | International Standard that sections must be transmitted in numerical order, but this is recommended, unless it is desired to transmit some sections of the table more frequently than others, e.g. due to random access considerations.

iv) **version_number**

When the characteristics of the Transport Stream described in the PSI change, (e.g. extra programs added, different composition of elementary streams for a given program), then new PSI data has to be sent with the updated information as the most recently transmitted version of the sections marked as "current" must always be valid. Decoders need to be able to identify whether the most recently received section is identical with the section they have already processed/stored (in which case the section can be discarded), or whether it is different, and may therefore signify a configuration change. This is achieved by sending a section with the same table_id, table_id_extension, and section_number as the previous section containing the relevant data, but with the next value version_number.

v) **current_next_indicator**

It is important to know at what point in the bitstream the PSI is valid. Each section can therefore be numbered as valid "now" (current), or as valid in the immediate future (next). This allows the transmission of a future configuration in advance of the change, giving the decoder the opportunity to prepare for the change. There is however no obligation to transmit the next version of a section in advance, but if it is transmitted, then it shall be the next correct version of that section.

C.3 The Mapping of Sections into Transport Stream Packets

Sections are mapped directly into Transport Stream packets, that is to say without a prior mapping into PES packets. Sections do not have to start at the beginning of Transport Stream packets, (although they may), because the start of the first section in the payload of a Transport Stream packet is pointed to by the pointer_field. The presence of the pointer_field is signalled by the payload_unit_start_indicator being set to a value of '1' in PSI packets. (In non-PSI packets, the indicator signals that a PES packet starts in the Transport Stream packet). The pointer_field points to the start of the first section in the Transport Stream packet. There is never more than one pointer_field in a Transport Stream packet, as the start of any other section can be identified by counting the length of the first and any subsequent sections, since no gaps between sections within a Transport Stream packet are allowed by the syntax.

It is important to note that within Transport Stream packets of any single PID value, one section must be finished before the next one is allowed to be started, or else it is not possible to identify to which section header the data belongs. If a section finishes before the end of a Transport Stream packet, but it is not convenient to open another section, a stuffing mechanism is provided to fill up the space. Stuffing is performed by filling each remaining byte of the packet with the value 0xFF. Consequently the table_id value 0xFF is forbidden, or else this would be confused with stuffing. Once a 0xFF byte has occurred at the end of a section, then the rest of the Transport Stream packet must be stuffed with 0xFF bytes, allowing a decoder to discard the rest of the Transport Stream packet. Stuffing can also be performed using the normal adaptation_field mechanism.

C.4 Repetition Rates and Random Access

In systems where random access is a consideration, it is recommended to re-transmit PSI sections several times, even when changes do not occur in the configuration, as in the general case, a decoder needs the PSI data to identify the contents of the Transport Stream, to be able to start decoding. This Recommendation | International Standard does not place any requirements on the repetition or occurrence rate of PSI sections. Clearly though, repeating sections frequently helps random access applications, whilst causing an increase in the amount of bitrate used by PSI data. If program mappings are static or quasi-static, they may be stored in the decoder to allow faster access to the data than having to wait for it to be re-transmitted. The trade-off between the amount of storage required and the desired impact on channel acquisition time may be made by the decoder manufacturer.

C.5 What is a Program?

The concept of a program has a precise definition within this Recommendation | International Standard [refer to 2.1.42 program (system)]. For a Transport Stream the time base is defined by the PCR. This effectively creates a virtual channel within the Transport Stream.

Note that this is not the same definition as is commonly used in broadcasting, where a "program" is a collection of elementary streams not only with a common timebase, but also with a common start and end time. A series of "broadcaster programs" (referred to in this annex as events) can be transmitted sequentially in a Transport Stream using the same program_number to create a "broadcasting conventional" TV-channel (sometimes called a service).

Event descriptions could be transmitted in private_sections().

A program is denoted by a program_number which has significance only within a Transport Stream. The program_number is a 16-bit unsigned integer and thus permits 65535 unique programs to exist within a Transport Stream (program_number 0 is reserved for identification of the NIT). Where several Transport Streams are available to the decoder (e.g. in a cable network), in order to successfully demultiplex a program, the decoder must be notified of both the transport_stream_id (to find the right multiplex) and the program_number of the service (to find the right program within the multiplex).

The Transport Stream mapping may be accomplished via the optional Network Information Table. Note that the Network Information Table may be stored in decoder non-volatile memory to reduce channel acquisition time. In this case, it needs to be transmitted only often enough to support timely decoder initialization set-up operations. The contents of the NIT are private, but shall take at least the minimum section structure.

C.6 Allocation of program_number

It may not be convenient in all cases to group together all the program element which share a common clock reference as one program. It is conceivable to have a multi-service Transport Stream with only one set of PCRs, common to all. In general, though, a broadcaster may prefer to logically split up the Transport Stream into several programs, where the PCR_PID (location of the clock reference) is always the same. This method of splitting the program elements into pseudo-independent programs can have several uses. Two examples follow:

i) *multilingual transmissions into separate markets*

One video stream may be accompanied by several audio streams in different languages. It is advisable to include an example of the ISO_639_language_descriptor associated with each audio stream to enable the selection of the correct program and audio. It is reasonable to have several program definitions with different program_numbers, where all the programs reference the same video stream and PCR_PID, but have different audio PIDs. It is, however, also reasonable and possible to list the video stream and all the audio streams as one program, where this does not exceed the section size limit of 1024 bytes.

ii) *Very large program definitions*

There is a maximum limit on the length of a section of 1024 bytes (including section header and CRC_32). This means that no single program definition may exceed this length. For the great majority of cases, even with each program element having several descriptors, this size is adequate. However, one may envisage cases in very high bitrate systems, which could exceed this limit. It is then in general possible to identify methods of splitting the references of the streams, so that they do not all have to be listed together. Some program elements could be referenced under more than one program, and some under only one or the other, but not both.

C.7 Usage of PSI in a Typical System

A communications system, especially in broadcast applications, may consist of many individual Transport Streams. Each one of the four PSI data structures may appear in each and every Transport Stream in a system. There must always be a complete version of the program association table listing all programs within the Transport Stream and a complete TS program map table, containing complete program definitions for all programs within the Transport Stream. If any streams are scrambled, then there must also be a conditional access table present listing the relevant Entitlement Management Messages (EMM) streams. The presence of a NIT is fully optional.

The PSI tables are mapped into Transport Stream packets via the section structure described above. Each section has a table_id field in its header, allowing sections from PSI tables and private data in private_sections to be mixed in Transport Stream packets of the same PID value or even in the same Transport Stream packet. Note, however, that within packets of the same PID, a complete section must be transmitted before the next section can be started. This is only possible for packets labelled as containing TS Program Map Table section or NIT packets however, since private sections may not be mapped into PAT or CAT packets.

It is required that all PAT sections be mapped into Transport Stream packets with PID = 0x0000 and all CA sections be mapped into packets with PID = 0x0001. PMT sections may be mapped into packets of user-selected PID value, listed as the PMT_PID for each program in the Program Association Table. Likewise, the PID for the NIT-bearing Transport Stream packets is user-selected, but must be pointed to by the entry "program_number == 0x00" in the PAT, if the NIT exists.

The contents of any CA parameter streams are entirely private, but EMMs and ECMs must also be sent in Transport Stream packets to be compliant with this Recommendation | International Standard.

Private data tables may be sent using the private_section() syntax. Such tables could be used for example in a broadcasting environment to describe a service, an upcoming event, broadcast schedules and related information.

C.8 The Relationships of PSI Structures

Figure C.1 shows an example of the relationship between the four PSI structures and the Transport Stream. Other examples are possible, but the figure shows the primary connections.

In the following subclauses, each PSI table is described.

C.8.1 Program Association Table

Every Transport Stream must contain a complete valid Program Association Table. The Program Association Table gives the correspondence between a program_number and the PID of the Transport Stream packets that carry the definition of that program (the PMT_PID). The PAT may be partitioned into up to 255 sections before it is mapped into Transport Stream packets. Each section carries a part of the overall PAT. This partitioning may be desirable to minimize data loss in error conditions. That is, packet loss or bit errors may be localized to smaller sections of the PAT, thus allowing other sections to still be received and correctly decoded. If all PAT information is put into one section, an error causing a changed bit in the table_id, for example, would cause the loss of the entire PAT. However, this is still permitted as long as the section does not extend beyond the 1024-byte maximum length limit.

Program 0 (zero) is reserved and is used to specify the Network PID. This is a pointer to the Transport Stream packets which carry the Network Information Table.

The Program Association Table is always transmitted without encryption.

C.8.2 Program Map Table

The Program Map Table provides the mapping between a program number and the program elements that comprise it. This table is present in Transport Stream packets having one or more privately-selected PID values. These Transport Stream packets may contain other private structures as defined by the table_id field. It is possible to have TS PMT sections referring to different programs carried in Transport Stream packets having a common PID value.

This Recommendation | International Standard requires a minimum of program identification: program number, PCR PID, stream types and program elements PIDs. Additional information for either programs or elementary streams may be conveyed by use of the descriptor() construct. Refer to C.8.6.

Private data may also be sent in Transport Stream packets denoted as carrying TS program map table sections. This is accomplished by the use of the private_section(). In a private_section() the application decides whether version_number and current_next_indicator represent the values of these fields for a single section or whether they are applicable to many sections as parts of a larger private table.

NOTE 1 – Transport stream packets containing the Program Map Table are transmitted unencrypted.

NOTE 2 – It is possible to transmit information on events in private descriptors carried within the TS_program_map_section()s.

C.8.3 Conditional Access Table

The Conditional Access (CA) Table gives the association between one or more CA systems, their EMM streams and any special parameters associated with them.

NOTE – The (private) contents of the Transport Stream packets containing EMM and CA parameters if present will, in general, be encrypted (scrambled).

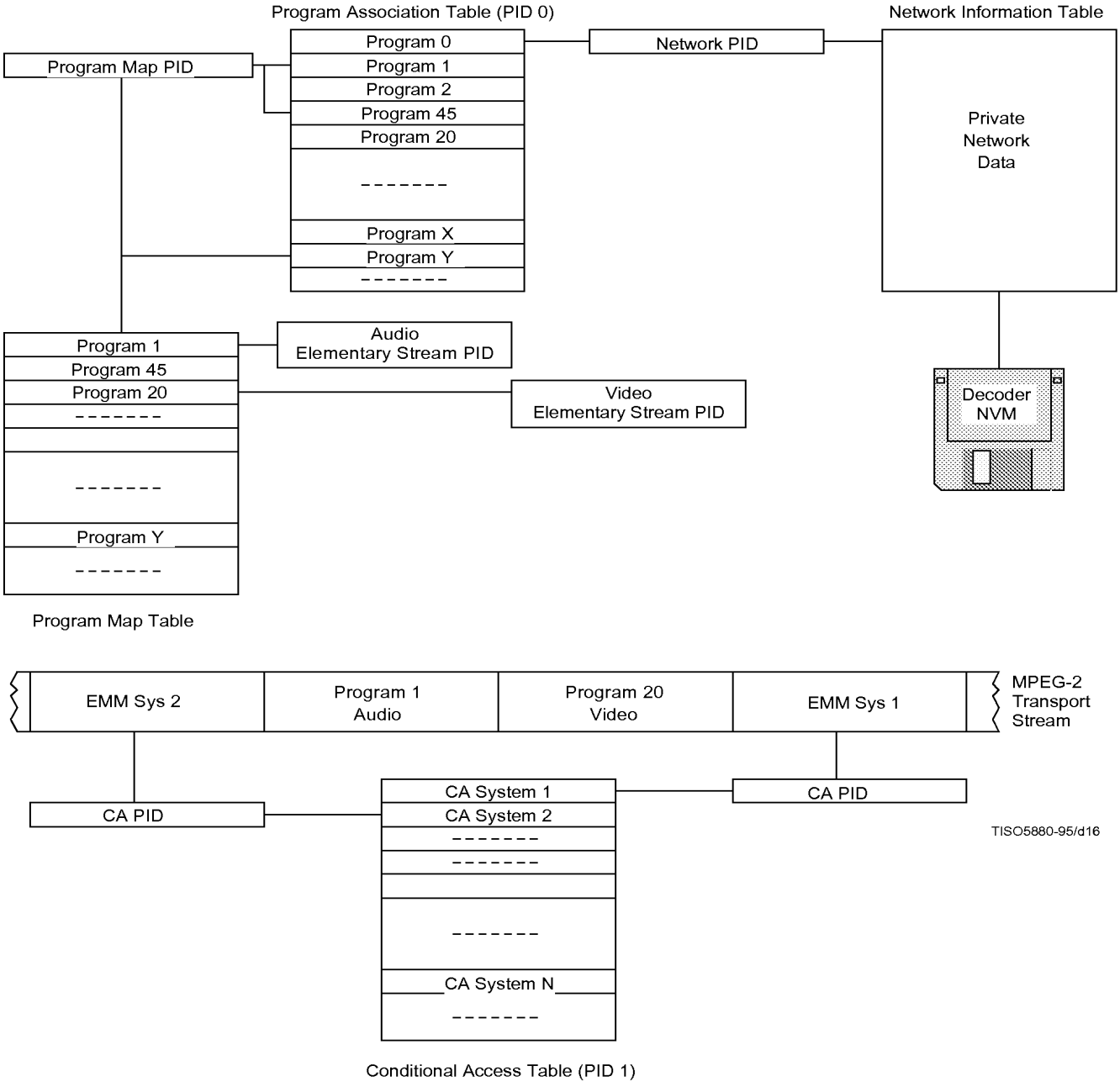


Figure C.1 – Program and network mapping relationships

C.8.4 Network Information Table

The contents of the NIT are private and not specified by this Recommendation | International Standard. In general, it will contain mappings of user-selected services with transport_stream_ids, channel frequencies, satellite transponder numbers, modulation characteristics, etc.

C.8.5 Private_section()

Private_sections() can occur in two basic forms, the short version (where only the fields up to and including section_length are included) or the long version (where all the fields up to and including last_section_number are present, and after the private data bytes the CRC_32 field is present).

Private_section()s can occur in PIDs which are labelled as PMT_PIDs or in Transport Stream packets with other PID values which contain exclusively private_sections(), including the PID allocated to the NIT. If the Transport Stream packets of the PID carrying the private_section()s are identified as a PID carrying private_sections (stream_type assignment value 0x05), then only private_sections may occur in Transport Stream packets of that PID value. The sections may be either of the short or long type.

C.8.6 Descriptors

There are several normative descriptors defined in this Recommendation | International Standard. Many more private descriptors may also be defined. All descriptors have a common format: {tag, length, data}. Any privately defined descriptors must adhere to this format. The data portion of these private descriptors are privately defined.

One descriptor (the CA_descriptor()), is used to indicate the location (PID value of transport packets) of ECM data associated with program elements when it is found in a TS PMT section. When found in a CA section it refers to EMMs.

In order to extend the number of private_descriptors available, the following mechanism could be used: A private descriptor_tag could be privately defined to be constructed as a composite descriptor. This entails privately defining a further sub_descriptor as the first field of the private data bytes of the private descriptor. The described structure is as indicated in Tables C.1 and C.2.

Table C.1 – Composite_descriptor

Syntax	No. of bits	Mnemonic
Composite_descriptor(){		
descriptor_tag(privately defined)	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++){		
sub_descriptor()		
}		
}		

Table C.2 – Sub-descriptor

Syntax	No. of bits	Mnemonic
sub_descriptor() {		
sub_descriptor_tag	**8**	**uimsbf**
sub_descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
private_data_byte	**8**	**uimsbf**
}		
}		

C.9 Bandwidth Utilization and Signal Acquisition Time

Any implementation of an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bitstream must make reasonable bandwidth demands for PSI information and, in applications where random access is a consideration, should promote fast signal acquisition. This subclause analyses this issue and gives some broadcast application examples.

The packet-based nature of the Transport Stream allows for the interspersing of PSI information with fine granularity in the multiplexed data. This provides significant flexibility in the construction and transmission of PSI.

Signal acquisition time in a real decoder is dependent on many factors, including: FDM tuning slew time, demultiplexing time, sequence headers, I-frame occurrence rate and scrambling key retrieval and processing.

This subclause examines both the bitrate and signal acquisition time impacts of the PSI syntax subclauses 2.4.4.4 and 2.4.4.9. It is assumed that the Conditional Access Table does not need to be received dynamically at every program change. This assumption is also made of the private EMM streams. This is because these streams do not contain the quickly-varying ECM components used for program element scrambling (encryption).

Also, in the discussion below, the time to acquire and process ECMs has been neglected.

Tables C.3 and C.4 provide bandwidth usage values for a range of Transport Stream conditions. One axis of the table is the number of programs contained in a single Transport Stream. The other axis is the frequency with which the PSI information is transmitted in the Transport Stream.

This frequency will be a key determinant of the component of signal acquisition time due to PSI structures.

Both bandwidth usage tables assume that only the minimum program mapping information is provided. This means that the PID values and stream types are provided with no additional descriptors. All programs in the example are composed of two elementary streams. Program associations are 2 bytes long, while the minimal program map is 26 bytes long. There is additional overhead associated with version numbers, section lengths, etc. This will be on the order of 1-3% of the total PSI bitrate usage in sections of moderate to maximum length (a few hundred bytes to 1024 bytes) and will thus be ignored here.

The above assumptions allow forty-six (46) program associations to map into one Program Association Table Transport Stream packet (if no adaptation field is present). Similarly, seven (7) TS_program_map_sections fit into a single Transport Stream packet. It may be noted that to facilitate easy "drop/add" it is possible to transmit only one (1) TS_program_map_section per PMT_PID. This may cause an undesirable increase in PSI bitrate usage, however.

Table C.3 – Program association table bandwidth usage (bit/s)

Number of Programs Per Transport Stream

		1	5	10	32	128
Frequency of PA Table Information (s^{-1})	1	1504	1504	1504	1504	4512
	10	15040	15040	15040	15040	45120
	25	37600	37600	37600	37600	112800
	50	75200	75200	75200	75200	225600
	100	150400	150400	150400	150400	451200
NOTE – Since 46 **program_association_sections** fit into one transport packet, the numbers in the table do not change until the last column.						

Table C.4 – Program map table bandwidth usage (bit/s)

Number of Programs Per Transport Stream

		1	5	10	32	128
Frequency of PM Table Information (s^{-1})	1	1504	1504	3008	7520	28576
	10	15040	15040	30080	75200	285760
	25	37600	37600	75200	188000	714400
	50	75200	75200	150400	376000	1428800
	100	150400	150400	300800	601600	2857600

Using a frequency of 25 Hz for the two PSI Tables, yields a worst case contribution to the signal acquisition time of approximately 80 ms. This would only occur when the required PAT data was "just missed" and then, once the PAT was acquired and decoded, the required PMT data was also "just missed". This doubling of the worst case acquisition time is one disadvantage of the extra level of indirection introduced by the PAT structure. This effect could be reduced by co-ordinated transmission of related PAT and PMT packets. Presumably, the advantage that this approach offers for "drop/add" re-multiplexing operations is compensatory.

With the 25 Hz PSI frequency, the following examples may be constructed (all examples leave ample allowance for various datalink, FEC, CA and routing overheads):

6 MHz CATV channel

- five 5.2-Mbit/s programs: 26.5 Mbit/s (includes transport overhead)
- total PSI bandwidth: 5.2 kbit/s
- CA bandwidth: 500 kbit/s

  total ITU-T Rec. H.222.0 | ISO/IEC 13818-1 transport bandwidth: 27.1 Mbit/s
- PSI Overhead: 0.28 %

OC-3 fiber channel (155 Mbit/s)

- 32 3.9-Mbit/s programs: 127.5 Mbit/s (includes transport overhead)
- total PSI bandwidth: 225.6 kbit/s
- CA bandwidth: 500 kbit/s

  total ITU-T Rec. H.222.0 | ISO/IEC 13818-1 transport bandwidth: 128.2 Mbit/s
- PSI Overhead: 0.18 %

C-band satellite transponder

- 128 256-kbit/s audio programs: 33.5 Mbit/s (includes transport overhead)
- total PSI bandwidth: 826.4 kbit/s
- CA bandwidth: 500 kbit/s

  total ITU-T Rec. H.222.0 | ISO/IEC 13818-1 transport bandwidth: 34.7 Mbit/s
- PSI Overhead: 2.4 % (actually would be lower if only one PID used per program)

As expected, the percent overhead increases for lower-rate services since many more services are possible per Transport Stream. However, the overhead is not excessive in all cases. Higher transmission rates (than 25 Hz) for the PSI data may be used to decrease the impact on channel acquisition time with only modest bitrate demand increases.

Annex D

Systems Timing Model and Application Implications of this Recommendation | International Standard

(This annex does not form an integral part of this Recommendation | International Standard)

D.0 Introduction

The ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Systems specification includes a specific timing model for the sampling, encoding, encoder buffering, transmission, reception, decoder buffering, decoding, and presentation of digital audio and video in combination. This model is embodied directly in the specification of the syntax and semantic requirements of compliant ITU-T Rec. H.222.0 | ISO/IEC 13818-1 data streams. Given that a decoding system receives a compliant bit stream that is delivered correctly in accordance with the timing model it is straightforward to implement the decoder such that it produces as output high quality audio and video which are properly synchronized. There is no normative requirement, however, that decoders be implemented in such a way as to provide such high quality presentation output. In applications where the data are not delivered to the decoder with correct timing, it may be possible to produce the desired presentation output, however such capabilities are not in general guaranteed. This informative annex describes the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Systems timing model in detail, and gives some suggestions for implementing decoder systems to suit some typical applications.

D.0.1 Timing Model

ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Systems embodies a timing model in which all digitized pictures and audio samples that enter the encoder are presented exactly once each, after a constant end to end delay, at the output of the decoder. As such, the sample rates, i.e. the video frame rate and the audio sample rate, are precisely the same at the decoder as they are at the encoder. This timing model is diagrammed in Figure D.1:

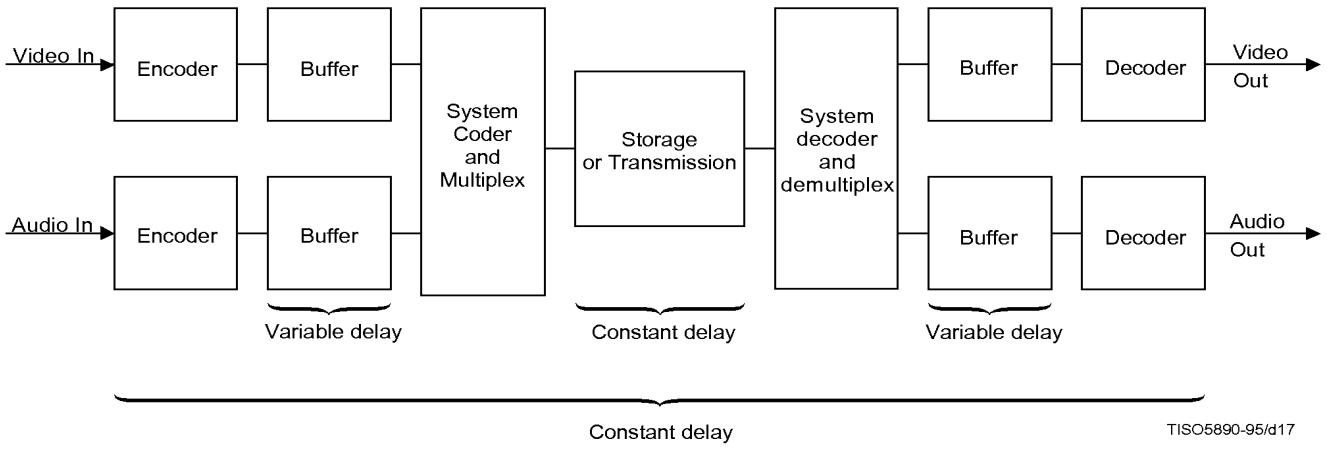


Figure D.1 – Constant delay model

As indicated in Figure D.1, the delay from the input to the encoder to the output or presentation from the decoder is constant in this model[1)], while the delay through each of the encoder and decoder buffers is variable. Not only is the delay through each of these buffers variable within the path of one elementary stream, the individual buffer delays in the video and audio paths differ as well. Therefore the relative location of coded bits representing audio or video in the combined stream does not indicate synchronization information. The relative location of coded audio and video is constrained only by the System Target Decoder (STD) model such that the decoder buffers must behave properly; therefore coded audio and video that represent sound and pictures that are to be presented simultaneously may be separated in time within the coded bit stream by as much as one second, which is the maximum decoder buffer delay that is allowed in the STD model.

1) Constant delay as indicated for the entire system is required for correct synchronizaton, however some deviations are possible. Network delay is discussed as being constant. Slight deviations may be tolerated, and network adaptation may allow greater variations of network delay. Both of these are discussed later.

The audio and video sample rates at the encoder are significantly different from one another, and may or may not have an exact and fixed relationship to one another, depending on whether the combined stream is a Program Stream or a Transport Stream, and on whether the System_audio_locked and System_video_locked flags are set in the Program Stream. The duration of a block of audio samples (an audio presentation unit) is generally not the same as the duration of a video picture.

There is a single, common system clock in the encoder, and this clock is used to create timestamps that indicate the correct presentation and decoding timing of audio and video, as well as to create timestamps that indicate the instantaneous values of the system clock itself at sampled intervals. The timestamps that indicate the presentation time of audio and video are called Presentation Time Stamps (PTS). Those that indicate the decoding time are called Decoding Timestamps (DTS), and those that indicate the value of the system clock are called the System Clock Reference (SCR) in Program Streams and the Program Clock Reference (PCR) in Transport Streams. It is the presence of this common system clock in the encoder, the timestamps that are created from it, and the recreation of the clock in the decoder and the correct use of the timestamps that provide the facility to synchronize properly the operation of the decoder.

Encoder implementations may not follow this model exactly, however the data stream which results from the actual encoder, storage system, network, and one or more multiplexor must follow the model precisely. (Delivery of the data may deviate somewhat, depending on the application). Therefore in this annex, the term "encoder system clock" is used to mean either the actual common system clock as described in this model or the equivalent function, however it may be implemented.

Since the end-to-end delay through the entire system is constant, the audio and video presentations are precisely synchronized. The construction of System bit streams is constrained such that when they are decoded by a decoder that follows this model with the appropriately sized decoder buffers, those buffers are guaranteed never to overflow nor underflow, with specific exceptions allowing intentional underflow.

In order for the decoder system to incur the precise amount of delay that causes the entire end-to-end delay to be constant, it is necessary for the decoder to have a system clock whose frequency of operation and absolute instantaneous value match those of the encoder. The information necessary to convey the encoder's system clock is encoded in the SCR or PCR; this function is explained below.

Decoders which are implemented in accordance with this timing model such that they present audio samples and video pictures exactly once (with specific intentionally coded exceptions), at a constant rate, and such that decoder buffers behave as in the model, are referred to in this annex as precisely timed decoders, or those that produce precisely timed output. Decoder implementations are not required by this International Standard to present audio and video in accordance with this model; it is possible to construct decoders that do not have constant delay, or equivalently do not present each picture or audio sample exactly once. In such implementations, however, the synchronization between presented audio and video may not be precise, and the behaviour of the decoder buffers may not follow the reference decoder model. It is important to avoid overflow at the decoder buffers, as overflow causes a loss of data that may have significant effects on the resulting decoding process. This annex covers primarily the operation of such precisely timed decoders and some of the options that are available in implementing these decoders.

D.0.2 Audio and Video Presentation Synchronization

Within the coding of this Recommendation | International Standard Systems data are timestamps concerning the presentation and decoding of video pictures and blocks of audio samples. The pictures and blocks are called "Presentation Units", abbreviated PU. The sets of coded bits which represent the PUs and which are included within the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 bit stream are called "Access Units", abbreviated AU. An audio access unit is abbreviated AAU, and a video access unit is abbreviated VAU. In ISO/IEC 13818-3 audio the term "audio frame" has the same meaning as AAU or APU (audio presentation unit) depending on the context. A video presentation unit (VPU) is a picture, and a VAU is a coded picture.

Some, but not necessarily all, AAUs and VAUs have associated with them PTSs. A PTS indicates the time that the PU which results from decoding the AU which is associated with the PTS should be presented to the user. The audio PTSs and video PTSs are both samples from a common time clock, which is referred to as the System Time Clock or STC. With the correct values of audio and video PTSs included in the data stream, and with the presentation of the audio and video PUs occurring at the time indicated by the appropriate PTSs in terms of the common STC, precise synchronization of the presented audio and video is achieved at the decoding system. While the STC is not part of the normative content of this Recommendation | International Standard, and the equivalent information is conveyed in this Recommendation | International Standard via such terms as the system_clock_frequency, the STC is an important and convenient element for explaining the timing model, and it is generally practical to implement encoders and decoders which include an STC in some form.

PTSs are required for the conveyance of accurate relative timing between audio and video, since the audio and video PUs generally have significantly different and essentially unrelated duration. For example, audio PUs of 1152 samples each at a sample rate of 44 100 samples per second have a duration of approximately 26,12 ms, and video PUs at a frame rate of 29,97 Hz have a duration of approximately 33,76 ms. In general the temporal boundaries of APUs and VPUs rarely, if ever, coincide. Separate PTSs for audio and video provide the information that indicates the precise temporal relation of audio and video PUs without requiring any specific relationship between the duration and interval of audio and video PUs.

The values of the PTS fields are defined in terms of the System Target Decoder or STD, which is a fundamental normative constraint on all System bit streams. The STD is a mathematical model of an idealized decoder which specifies precisely the movement of all bits into and out of the decoder's buffers, and the basic semantic constraint imposed on the bit stream is that the buffers within the STD must never overflow nor underflow, with specific exceptions provided for underflow in special cases. In the STD model the virtual decoder is always exactly synchronized with the data source, and audio and video decoding and presentation are exactly synchronized. While exact and consistent, the STD is somewhat simplified with respect to physical implementations of decoders in order to clarify its specification and to facilitate its broad application to a variety of decoder implementations. In particular, in the STD model each of the operations performed on the bit stream in the decoder is performed instantaneously, with the obvious exception of the time that bits spend in the decoder buffers. In a real decoder system the individual audio and video decoders do not perform instantaneously, and their delays must be taken into account in the design of the implementation. For example, if video pictures are decoded in exactly one picture presentation interval 1/P, where P is the frame rate, and compressed video data are arriving at the decoder at bit rate R, the completion of removing bits associated with each picture is delayed from the time indicated in the PTS and DTS fields by 1/P, and the video decoder buffer must be larger than that specified in the STD model by R/P. The video presentation is likewise delayed with respect to the STD, and the PTS should be handled accordingly. Since the video is delayed, the audio decoding and presentation should be delayed by a similar amount in order to provide correct synchronization. Delaying decoding and presentation of audio and video in a decoder may be implemented for example by adding a constant to the PTS values when they are used within the decoder.

Another difference between the STD and precise practical decoder implementation is that in the STD model the explicit assumption is made that the final audio and video output is presented to the user instantaneously and without further delay. This may not be the case in practice, particularly with cathode-ray tube displays, and this additional delay should also be taken into account in the design. Encoders are required to encode audio and video such that the correct synchronization is achieved when the data is decoded with the STD. Delays in the input and sampling of audio and video, such as video camera optical charge integration, must be taken into account in the encoder.

In the STD model proper synchronization is assumed and the timestamps and buffer behaviour are tested against this assumption as a condition of bit stream validity. Of course in a physical decoder precise synchronization is not automatically the case, particularly upon start-up and in the presence of timing jitter. Precise decoder timing is a goal to be targeted by decoder designs. Inaccuracy in decoder timing affects the behaviour of the decoder buffers. These topics are covered in more detail in later subclauses of this annex.

The STD includes Decoding Time Stamps (DTS) as well as PTS fields. The DTS refers to the time that an AU is to be extracted from the decoder buffer and decoded in the STD model. Since the audio and video elementary stream decoders are instantaneous in the STD, the decoding time and presentation time are identical in most cases; the only exception occurs with video pictures which have undergone re-ordering within the coded bit stream, i.e. I- and P-pictures in the case of non-low-delay video sequences. In cases where re-ordering exists, a temporary delay buffer in the video decoder is used to store the appropriate decoded I- or P-picture until it should be presented. In all cases where the decoding and presentation times are identical in the STD, i.e. all AAUs, B-picture VAUs, and I- and P-picture VAUs within low-delay video sequences, the DTS is not coded, as it would have the same value as the PTS. Where the values differ, both are coded if either is coded. For all AUs where only the PTS is coded, this field may be interpreted as being both the PTS and the DTS.

Since PTS and DTS values are not required for every AAU and VAU, the decoder may choose to interpolate values which are not coded. PTS values are required with intervals not exceeding 700 ms in each elementary audio and video stream. These time intervals are measured in presentation time, that is, in the same context as the values of the fields, not in terms of the times that the fields are transmitted and received. In cases of data streams where the system, video and audio clocks are locked, as defined in the normative part of this Recommendation | International Standard, each AU following one for which a DTS or PTS is explicitly coded has an effective decoding time of the sum of that for the previous AU plus a fixed and specified difference in value of the STC. For example, in video coded at 29,97 Hz each picture has a difference in time of 3003 cycles of the 90 kHz portion of the STC from the previous picture when the video and system clocks are locked. The same time relationship exists for decoding successive AUs, although re-ordering delay in the decoder affects the relationship between decoder AUs and presented PUs. When the data stream is coded such that the video or audio clock is not locked to the system clock the time difference between decoding successive AUs may be estimated using the same values as indicated above; however these time differences are not exact due to the fact that relationships between the frame rate, audio sample rate, and system clock frequency were not exact at the encoder.

Note that the PTS and DTS fields do not, by themselves, indicate the correct fullness of the decoder buffers at start up nor at any other time, and equivalently, they do not indicate the amount of time delay that should elapse upon receiving the initial bits of a data stream before decoding should start. This information is retrieved by combining the functions of the PTS and DTS fields and correct clock recovery, which is covered below. In the STD model, and therefore in decoders which are modelled after it, the decoder buffer behaviour is determined completely by the SCR (or PCR) values, the times that they are received, and the PTS and DTS values, assuming that data is delivered in accordance with the timing model. This information specifies the time that coded data spends in the decoder buffers. The amount of data that is in the coded data buffers is not explicitly specified, and this information is not necessary, since the timing is fully specified. Note also that the fullness of the data buffers may vary considerably with time in a fashion that is not predictable by the decoder, except through the proper use of the timestamps.

In order for the audio and video PTSs to refer correctly to a common STC, a correctly timed common clock must be made available within the decoder system. This is subject of the next subclause.

D.0.3 System Time Clock recovery in the decoder

Within the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Systems data stream there are, in addition to the PTS and DTS fields, clock reference timestamps. These references are samples of the system time clock, which are applicable both to a decoder and to an encoder. They have a resolution of one part in 27 000 000 per second, and occur at intervals up to 100 ms in Transport Streams, or up to 700 ms in Program Streams. As such, they can be utilized to implement clock reconstruction control loops in decoders with sufficient accuracy for all identified applications.

In the Program Stream, the clock reference field is called the System Clock Reference or SCR. In the Transport Stream, the clock reference field is called the Program Clock Reference or PCR. In general the SCR and PCR definitions may be considered to be equivalent, although there are distinctions. The remainder of this subclause uses the term SCR for clarity; the same statements apply to the PCR except where otherwise noted. The PCR in Transport Streams provides the clock reference for one program, where a program is a set of elementary streams that have a common time base and are intended for synchronized decoding and presentation. There may be multiple programs in one Transport Stream, and each may have an independent time base and a separate set of PCRs.

The SCR field indicates the correct value of the STC when the SCR is received at the decoder. Since the SCR occupies more than one byte of data, and System data streams are defined as streams of bytes, the SCR is defined to arrive at the decoder when the last byte of the system_clock_reference_base field is received at the decoder. Alternatively the SCR can be interpreted as the time that the SCR field should arrive at the decoder, assuming that the STC is already known to be correct. Which interpretation is used depends on the structure of the application system. In applications where the data source can be controlled by the decoder, such as a locally attached DSM, it is possible for the decoder to have an autonomous STC frequency, and so the STC need not be recovered. In many important applications, however, this assumption cannot be made correctly. For example, consider the case where a data stream is delivered simultaneously to multiple decoders. If each decoder has its own autonomous STC with its own independent clock frequency, the SCRs cannot be assured to arrive at the correct time at all decoders; one decoder will in general require the SCRs sooner than the source is delivering them, while another requires them later. This difference cannot be made up with a finite size data buffer over an unbounded length of time of data reception. Therefore the following addresses primarily the case where the STC must slave its timing to the received SCRs (or PCRs).

In a correctly constructed and delivered ITU-T Rec. H.222.0 | ISO/IEC 13818-1 data stream, each SCR arrives at the decoder at precisely the time indicated by the value of that SCR. In this context, "time" means correct value of the STC. In concept, this STC value is the same value that the encoder's STC had when the SCR was stored or transmitted. However, the encoding may have been performed not in real time or the data stream may have been modified since it was originally encoded, and in general the encoder or data source may be implemented in a variety of ways such that the encoder's STC may be a theoretical quantity.

If the decoder's clock frequency matches exactly that of the encoder, then the decoding and presentation of video and audio will automatically have the same rate as those at the encoder, and the end-to-end delay will be constant. With matched encoder and decoder clock frequencies, any correct SCR value can be used to set the instantaneous value of the decoder's STC, and from that time on the decoder's STC will match that of the encoder without the need for further adjustment. This condition remains true until there is a discontinuity of timing, such as the end of a Program Stream or the presence of a discontinuity indicator in a Transport Stream.

In practice a decoder's free-running system clock frequency will not match the encoder's system clock frequency which is sampled and indicated in the SCR values. The decoder's STC can be made to slave its timing to the encoder using the received SCRs. The prototypical method of slaving the decoder's clock to the received data stream is via a phase-locked loop (PLL). Variations of a basic PLL, or other methods, may be appropriate, depending on the specific application requirements.

A straight-forward PLL which recovers the STC in a decoder is diagrammed and described here.

Figure D.2 shows a classic PLL, except that the reference and feedback terms are numbers (STC and SCR or PCR values) instead of signal events such as edges.

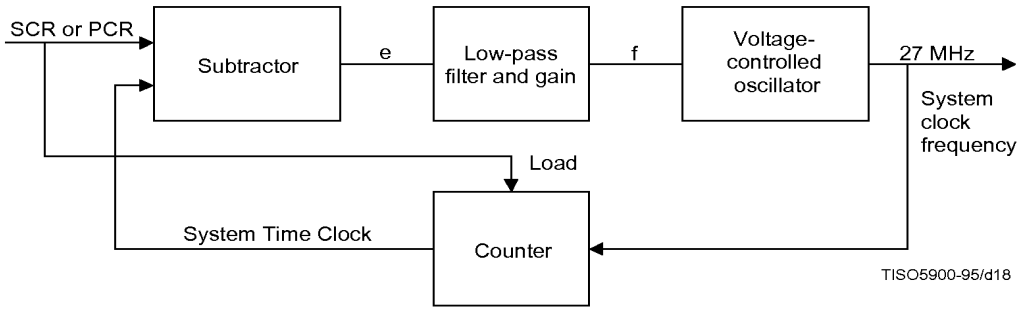


Figure D.2 – STC recovery using PLL

Upon initial acquisition of a new time base, i.e. a new program, the STC is set to the current value encoded in the SCRs. Typically the first SCR is loaded directly into the STC counter, and the PLL is subsequently operated as a closed loop. Variations on this method may be appropriate, i.e. if the values of the SCRs are suspect due to jitter or errors.

The closed-loop action of the PLL is as follows. At the moment that each SCR (or PCR) arrives at the decoder, that value is compared with the current value of the STC. The difference is a number, which has one part in units of 90 kHz and one part in terms of 300 times this frequency, i.e. 27 MHz. The difference value is linearized to be in a single number space, typically units of 27 MHz, and is called "**e**", the error term in the loop. The sequence of **e** terms is input to the low-pass filter and gain stage, which are designed according to the requirements of the application. The output of this stage is a control signal "**f**" which controls the instantaneous frequency of the Voltage Controlled Oscillator (VCO). The output of the VCO is an oscillator signal with a nominal frequency of 27 MHz; this signal is used as the system clock frequency within the decoder. The 27 MHz clock is input to a counter which produces the current STC values, which consist of both a 27 MHz extension, produced by dividing by 300, and a 90 kHz base value which is derived by counting the 90 kHz results in a 33-bit counter. The 33 bit, 90 kHz portion of the STC output is used as needed for comparison with PTS and DTS values. The complete STC is also the feedback input to the subtractor.

The bounded maximum interval between successive SCRs (700 ms) or PCRs (100 ms) allows the design and construction of PLLs which are known to be stable. The bandwidth of the PLLs has an upper bound imposed by this interval. As shown below, in many applications the PLL required has a very low bandwidth, and so this bound typically does not impose a significant limitation on the decoder design and performance.

If the free-running or initial frequency of the VCO is close enough to the correct, encoder's system clock frequency, the decoder may be able to operate satisfactorily as soon as the STC is initialized correctly, before the PLL has reached a defined locked state. For a given decoder STC frequency which differs by a bounded amount from the frequency encoded in the SCRs and which is within the absolute frequency bounds required by the decoder application, the effect of the mis-match between the encoder's and the decoder's STC frequencies if there were not PLL is the gradual and unavoidable increase or decrease of the fullness of the decoder's buffers, such that overflow or underflow would occur eventually with any finite size of decoder buffers. Therefore the amount of time allowable before the decoder's STC frequency is locked to that of the encoder is determined by the allowable amount of additional decoder buffer size and delay.

If the SCRs are received by the decoder with values and timing that reflect instantaneously correct samples of a constant frequency STC in the encoder, then the error term **e** converges to an essentially constant value after the loop has reached the locked state. This condition of correct SCR values is synonymous with either constant-delay storage and transmission of the data from the encoder to the decoder, or if this delay is not constant, the effective equivalent of constant delay storage and transmission with the SCR values having been corrected to reflect the variations in delay. With the values of **e** converging to a constant, variations in the instantaneous VCO frequency become essentially zero after the loop is locked; the VCO is said to have very little jitter or frequency slew. While the loop is in the process of locking, the rate of change of the VCO frequency, the frequency slew rate, can be controlled strictly by the design of the low pass filter and gain stage. In general the VCO slew rate can be designed to meet application requirements, subject to constraints of decoder buffer size and delay.

D.0.4 SCR and PCR Jitter

If a network or a Transport Stream re-multiplexor varies the delay in delivering the data stream from the encoder or storage system to the decoder, such variations tend to cause a difference between the values of the SCRs (or PCRs) and the values that they should have when they are actually received. This is referred to as SCR or PCR jitter. For example, if the delay in delivering one SCR is greater than the delay experienced by other similar fields in the same program, that SCR is late. Similarly, if the delay is less than for other clock reference fields in the program, the field is early.

Timing jitter at the input to a decoder is reflected in the combination of the values of the SCRs and the times when they are received. Assuming a clock recovery structure as illustrated in Figure D.2, any such timing jitter will be reflected in the values of the error term **e**; and non-zero values of **e** induce variations in the values of **f**, resulting in variations in the frequency of the 27 MHz system clock. Variations in the frequency of the recovered clock may or may not be acceptable within decoder systems, depending on the specific application requirements. For example, in precisely timed decoders that produce composite video output, the recovered clock frequency is typically used to generate the composite video sample clock and the chroma sub-carrier; the applicable specifications for sub-carrier frequency stability may permit only very slow adjustment of the system clock frequency. In applications where a significant amount of SCR or PCR jitter is present at the decoder input and there are tight constraints on the frequency slew rate of the STC, the constraints of reasonable additional decoder buffer size and delay may not allow proper operation.

The presence of SCR or PCR jitter may be caused for example by network transmission which incorporates packet or cell multiplexing or variable delay of packets through the network, as may be caused by queuing delays or by variable network access time in shared-media systems.

Multiplexing or re-multiplexing of Transport or Program Streams changes the order and relative temporal location of data packets and therefore also of SCRs or PCRs. The change in temporal location of SCRs causes the value of previously correct SCRs to become incorrect, since in general the time at which they are delivered via a constant delay network is not correctly represented by their values. Similarly, a Program Stream or Transport Stream with correct SCRs or PCRs may be delivered over a network which imposes a variable delay on the data stream, without correcting the SCR or PCR values. The effect is once again SCR or PCR jitter, with attendant effects on the decoder design and performance. The worst case amount of jitter which is imposed by a network on the SCRs or PCRs received at a decoder depends on a number of factors which are beyond the scope of this Recommendation | International Standard, including the depth of queues implemented in each of the network switches and the total number of network switches or re-multiplexing operations which operate in cascade on the data stream.

In the case of a Transport Stream, correction of PCRs is necessary in a re-multiplex operation, creating a new Transport Stream from one or more Transport Streams. This correction is accomplished by adding a correction term to the PCR; this term can be computed as:

$$\Delta PCR = del_{act} - del_{const}$$

where del_{act} is the actual delay experienced by the PCR, and del_{const} is a constant which is used for all PCRs of that program. The value which should be used for del_{const} will depend on the strategy used by the original encoder/multiplexor. This strategy could be, for instance, to schedule packets as early as possible, in order to allow later transmission links to delay them. In Table D.1, three different multiplex strategies are shown together with the appropriate value for del_{const}.

Table D.1 – Re-multiplexing strategy

Strategy	del_{const}
Early	del_{min}
Late	del_{max}
Middle	del_{avg}

When designing a system, private agreements may be needed as to what strategy should be used by the encoder/multiplexors, since this will have an effect on the ability to perform any additional re-multiplexing.

The amount of multiplex jitter allowed is not normatively bounded in this Recommendation | International Standard. However, 4 ms is intended to be the maximum amount of jitter in a well behaved system.

In systems which include re-multiplexors special care might be necessary to ensure that the information in the Transport Stream is consistent. In particular, this applies to PSI and to discontinuity points. Changes in PSI tables might need to be inserted into a Transport Stream in such a way that subsequent re-multiplexor steps never move them so far that information becomes incorrect. For instance, a new version of PMT section in some cases should not be sent within 4 ms of the data affected by the change.

Similarly, it may be necessary for an encoder/mux to avoid inserting PTS or DTS in a ±4 ms window around a discontinuity point.

D.0.5 Clock Recovery in the Presence of Network Jitter

In applications in which there is any significant amount of jitter present in the received clock reference timestamps, there are several choices available for decoder designs; how the decoder is designed depends in large part on the requirements for the decoder's output signal characteristics as well as the characteristics of the input data and jitter.

Decoders in various applications may have differing requirements for the accuracy and stability of the recovered system clock, and the degree of this stability and accuracy that is required may be considered to fall along a single axis. One extreme of this axis may be considered to be those applications where the reconstructed system clock is used directly to synthesize a chroma sub-carrier for use in composite video. This requirement generally exists where the presented video is of the precisely timed type, as described above, such that each coded picture is presented exactly once, and where the output is composite video in compliance with the applicable specifications. In that case the chroma sub-carrier, the pixel clock, and the frame rate all have exactly specified ratios, and all of these have a defined relationship to the system clock. The composite video sub-carrier must have at least sufficient accuracy and stability that any normal television receiver's chroma sub-carrier PLL can lock to the sub-carrier, and the chroma signals which are demodulated using the recovered sub-carrier do not show visible chrominance phase artifacts. The requirement in some applications is to use the system clock to generate a sub-carrier that is in full compliance with the NTSC, PAL, or SECAM specifications, which are typically even more stringent than those imposed by typical television receivers. For example, the SMPTE specification for NTSC requires a sub-carrier accuracy of 3 ppm, with a maximum short term jitter of 1 ns per horizontal line time and a maximum long term drift of 0,1 Hz per second.

In applications where the recovered system clock is not used to generate a chroma sub-carrier, it may still be used to generate a pixel clock for video and it may be used to generate a sample clock for audio. These clocks have their own stability requirements that depend on the assumptions made about the receiving display monitor and on the acceptable amount of audio frequency drift, or "wow and flutter", at the decoder's output.

In applications where each picture and each audio sample are not presented exactly once, i.e. picture and audio sample "slipping" is allowed, the system clock may have relatively loose accuracy and stability requirements. This type of decoder may not have precise audio-video presentation synchronization, and the resulting audio and video presentation may not have the same quality as for precisely timed decoders.

The choice of requirements for the accuracy and stability of the recovered system clock is application dependent. The following focuses on the most stringent requirement which is identified above, i.e. where the system clock is to be used to generate a chroma sub-carrier.

D.0.6 System clock used for chroma sub-carrier generation

The decoder design requirements can be determined from the requirements on the resulting sub-carrier and the maximum amount of network jitter that must be accepted. Similarly, if the system clock performance requirements and the decoder design's capabilities are known, the tolerable maximum network jitter can be determined. While it is beyond the scope of this Recommendation | International Standard to state such requirements, the numbers which are needed to specify the design are identified in order to clarify the statement of the problem and to illustrate a representative design approach.

With a clock recovery PLL circuit as illustrated in Figure D.2, the recovered system clock must meet the requirements of a worst case frequency deviation from the nominal, measured in units of ppm (parts per million), and a worst case frequency slew rate, measured in ppm/s (ppm per second). The peak-to-peak uncorrected network timing jitter has a value that may be specified in milliseconds. In such a PLL the network timing jitter appears as the error term **e** in the diagram, and since the PLL acts as a low-pass filter on jitter at its input, the worst case effect on the 27 MHz output frequency occurs when there is a maximum amplitude step function of PCR timing at the input. The value **e** then has a maximum amplitude equal to the peak-to-peak jitter, which is represented numerically as the jitter times 2**33 in the base portion of the SCR or PCR encoding. The maximum rate of change of the output of the low pass filter (LPF), **f**, with this maximum value of **e** at its input, directly determines the maximum frequency slew rate of the 27 MHz output. For any given maximum value of **e** and maximum rate of change of **f** a LPF can be specified. However, as the gain or cut-off frequency of the LPF is reduced, the time required for the PLL to lock to the frequency represented by the SCRs or PCRs is increased. Implementation of PLLs with very long time constants can be achieved through the use of digital LPF techniques, and possibly analogue filter techniques. With digital LPF implementations, when the frequency term **f** is the input to an analogue VCO, **f** is quantized by a digital to analogue converter, whose step size should be considered when calculating the maximum slew rate of the output frequency.

In order to ensure that **e** converges to a value that approaches zero, the open loop gain of the PLL must be very high, such as might be implemented in an integrator function in the low-pass filter in the PLL.

With a given accuracy requirement, it may be reasonable to construct the PLL such that the initial operating frequency of the PLL meets the accuracy requirement. In this case the initial 27 MHz frequency before the PLL is locked is sufficiently accurate to meet the stated output frequency requirement. If it were not for the fact that the decoder's buffers would eventually overflow or underflow, this initial system clock frequency would be sufficient for long term operation. However, from the time the decoder begins to receive and decode data until the system clock is locked to the time and clock frequency that is represented by the received SCRs or PCRs, data is arriving at the buffers at a different rate than it is being extracted, or equivalently the decoder is extracting access units at times that differ from those of the System Target Decoder (STD) model. The decoder buffers will continue to become more or less full than those of the STD according to the trajectory of recovered system clock frequency with respect to the encoder's clock frequency. Depending on the relative initial VCO frequency and encoder system clock frequency, decoder buffer fullness is either increasing or decreasing. Assuming this relationship is not known, the decoder needs additional data buffering to allow for either case. The decoder should be constructed to delay all decoding operations by an amount of time that is at least equal to the amount of time that is represented by the additional buffering that is allocated for the case of the initial VCO frequency being greater than the encoder's clock frequency, in order to prevent buffer underflow. If the initial VCO frequency is not sufficiently accurate to meet the stated accuracy requirements, then the PLL must reach the locked state before decoding may begin, and there is a different set of considerations regarding the PLL behaviour during this time and the amount of additional buffering and static delay which is appropriate.

A step function in the input timing jitter which produces a step function in the error term **e** of the PLL in Figure D.2 must produce an output frequency term **f** such that when it is multiplied by the VCO gain the maximum rate of change is less than the specified frequency slew rate. The gain of the VCO is stated in terms of the amount of the change in output frequency with respect to a change in control input. An additional constraint on the LPF in the PLL is that the static value of **e** when the loop is locked must be bounded in order to bound the amount of additional buffering and static decoding delay that must be implemented. This term is minimized when the LPF has very high DC gain.

Clock recovery circuits which differ somewhat from that shown in Figure D.2 may be practical. For example, it may be possible to implement a control loop with a Numerically Controlled Oscillator (NCO) instead of a VCO, wherein the NCO uses a fixed frequency oscillator and clock cycles are inserted or deleted from normally periodic events at the output in order to adjust the decoding and presentation timing. There may be some difficulties with this type of approach when used with composite video, as there is a tendency to cause either problematic phase shifts of the sub-carrier or jitter in the horizontal or vertical scan timing. One possible approach is to adjust the period of horizontal scans at the start of vertical blanking, while maintaining the phase of the chroma sub-carrier.

In summary, depending on the values specified for the requirements, it may or may not be practical to construct a decoder which reconstructs the system clock with sufficient accuracy and stability, while maintaining desired decoder buffer sizes and added decoding delay.

D.0.7 Component video and audio reconstruction

If component video is produced at the decoder output, the requirements for timing accuracy and stability are generally less stringent than is the case for composite video. Typically the frequency tolerance is that which the display deflection circuitry can accept, and the stability tolerance is determined by the need to avoid visible image displacement on the display.

The same principles as illustrated above apply, however the specific requirements are generally easier to meet.

Audio sample rate reconstruction again follows the same principles, however the stability requirement is determined by the amount of acceptable long and short term sample rate variation. Using a PLL approach as illustrated in the previous subclause, short term deviation can be made to be very small, and longer term frequency variation is manifested as variation in perceived pitch. Again, once specified bounds on this variation are set specific design requirements can be determined.

D.0.8 Frame Slipping

In some applications where precise decoder timing is not required, the decoder's system time clock may not adjust its operating frequency to match the frequency represented by received SCRs (or PCRs); it may have a free-running 27 MHz clock instead, while still slaving the decoder's STC to the received data. In this case the STC value must be updated as needed to match the received SCRs. Updating the STC upon receipt of SCRs causes discontinuities in the STC value. The magnitude of these discontinuities depends upon the difference between the decoder's 27 MHz frequency and the encoder's 27 MHz, i.e. that which is represented by the received SCRs, and upon the time interval between successive received SCRs or PCRs. Since the decoder's 27 MHz system clock frequency is not locked to that of

the received data, it cannot be used to generate the video or audio sample clocks while maintaining the precise timing assumptions of presenting each video and audio presentation unit exactly once and of maintaining the same picture and audio presentation rate at the decoder and the encoder, with precise audio and video synchronization. There are multiple possibilities for implementing decoding and presentation systems using this structure.

In one type of implementation, the pictures and audio samples are decoded at the time indicated by the decoder's STC, while they are presented at slightly different times, according to the locally produced sample clocks. Depending on the relationships of the decoder's sample clocks to the encoder's system clock, pictures and audio samples may on occasion be presented more than one each or not at all; this is referred to as "frame slipping" or "sample slipping", in the case of audio. There may be perceptible artifacts introduced by this mechanism. The audio-video synchronization will in general not be precise, due to the units of time over which pictures, and perhaps audio presentation units, are repeated or deleted. Depending on the specific implementation, additional buffering in the decoder is generally needed for coded data or decoded presentation data. Decoding may be performed immediately before presentation, and not quite at the time indicated in the decoder's STC, or decoded presentation units may be stored for delayed and possibly repeated presentation. If decoding is performed at the time of presentation, a mechanism is required to support deleting the presentation of pictures and audio samples without causing problems in the decoding of predictively coded data.

D.0.9 Smoothing of network jitter

In some applications it may be possible to introduce a mechanism between a network and a decoder in order to reduce the degree of jitter which is introduced by a network. Whether such an approach is feasible depends on the type of streams received and the amount and type of jitter which is expected.

Both the Transport Stream and the Program Stream indicate within their syntax the rate at which the stream is intended to be input to a decoder. These indicated rates are not precise, and cannot be used to reconstruct data stream timing exactly. They may, however, be useful as part of a smoothing mechanism.

For example, a Transport Stream may be received from a network such that the data is delivered in bursts. It is possible to buffer the received data and to transmit data from the buffer to the decoder at an approximately constant rate such that the buffer remains approximately one-half full.

However, a variable rate stream should not be delivered at constant rate, and with variable rate streams the smoothing buffer should not always be one-half full. A constant average delay through the buffer requires a buffer fullness that varies with the data rate. The rate that data should be extracted from the buffer and input to the decoder can be approximated using the rate information present in the data stream. In Transport Streams the intended rate is determined by the values of the PCR fields and the number of Transport Stream bytes between them. In Program Streams the intended rate is explicitly specified as the Program_mux_rate, although as specified in this Recommendation | International Standard the rate may drop to zero at SCR locations, i.e. if the SCR arrives before the time expected when the data is delivered at the indicated rate.

In the case of variable rate streams, the correct fullness of the smoothing buffer varies with time, and may not be determined exactly from the rate information. In an alternative approach, the SCRs or PCRs may be used to measure the time when data enter the buffer and to control the time when data leave the buffer. A control loop can be designed to provide constant average delay through the buffer. It may be observed that such a design is similar to the control loop illustrated in Figure D.2. The performance obtainable from inserting such a smoothing mechanism before a decoder can also be achieved by cascading multiple clock recovery PLLs. The rejection of jitter from the received timing will benefit from the combined low pass filter effect of the cascaded PLLs.

Annex E

Data Transmission Applications

(This annex does not form an integral part of this Recommendation | International Standard)

E.0 General considerations

- ITU-T Rec. H.222.0 | ISO/IEC 13818-1 transport multiplex will be used to transmit data as well as video and audio.
- Data elementary streams are not continuous as may appear video and audio streams in broadcast applications.
- While it is already possible to identify the beginning of a PES packet, it is not always possible to identify the end of a PES packet by the beginning of the next PES packet, because it is possible for one or more Transport packet carrying PES packets to be lost.

E.1 Suggestion

A suitable solution is to transmit the following PES packet just after an associated PES packet. A PES packet without payload may be sent when there are no further PES packets to send.

Table E.1 is an example of such a PES packet.

Table E.1 – PES packet header example

PES packet header fields	Values
packet_start_code_prefix	0x000001
stream_id	assigned
PES_packet_length	0x0003
'10'	'10'
PES_scrambling_control	'00'
PES_priority	'0'
data_alignment_indicator	'0'
copyright	'0'
original_or_copy	'0'
PTS_DTS_flags	'00'
ESCR_flag	'0'
ES_rate_flag	'0'
DSM_trick_mode_flag	'0'
additonal_copy_info_flag	'0'
PES_CRC_flag	'0'
PES_extension_flag	'0'
PES_header_data_length	0x00

Annex F

Graphics of Syntax for this Recommendation | International Standard

(This annex does not form an integral part of this Recommendation | International Standard)

F.0 Introduction

This annex is an informative annex presenting graphically the Transport Stream and Program Stream syntax. This annex in no way replaces any normative clause(s).

In order to produce clear drawings, not all fields have been fully described or represented. Reserved fields may be omitted or indicated by areas with no detail. Fields length are indicated in bits.

F.0.1 Transport Stream syntax

See Figure F.1.

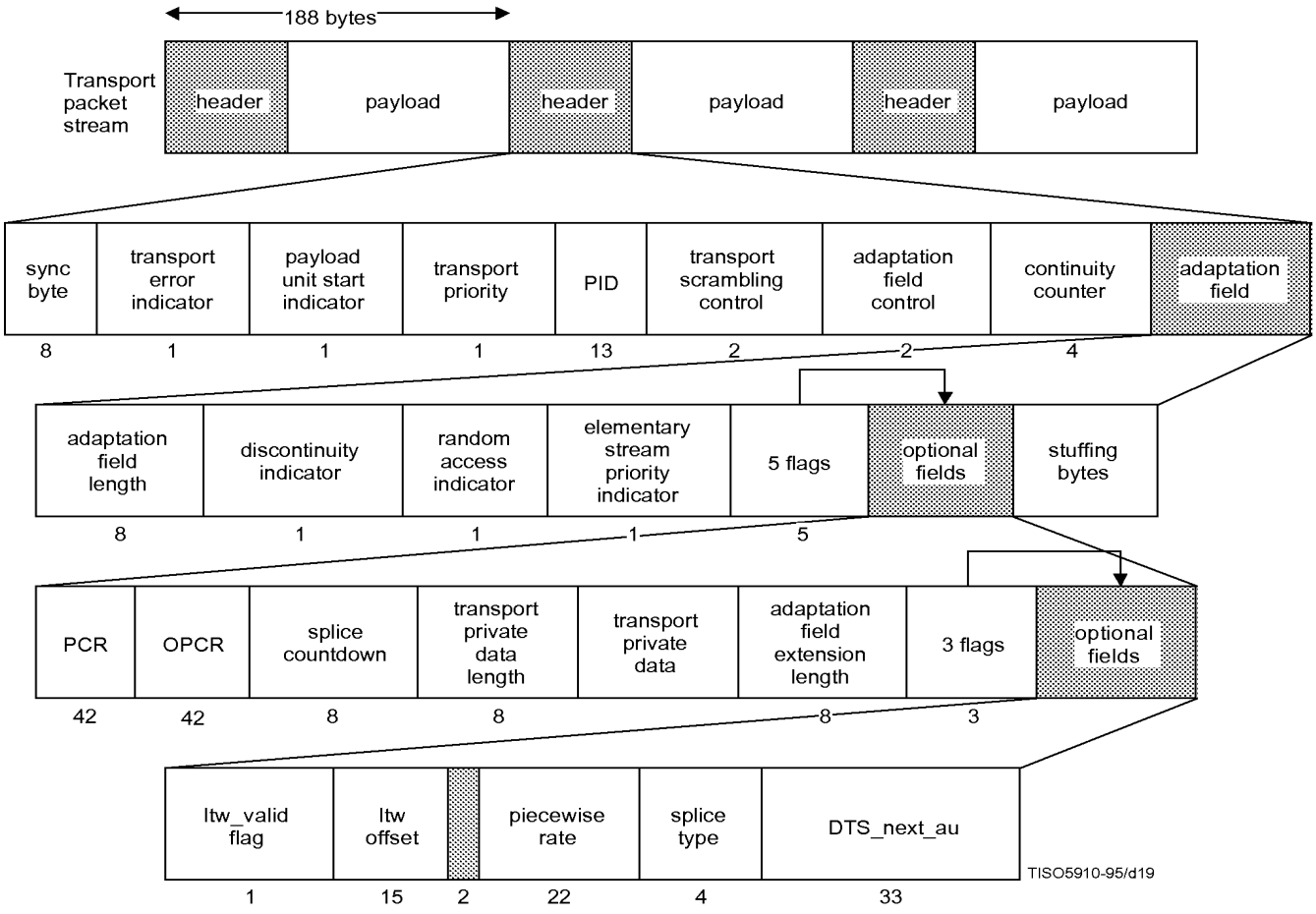


Figure F.1 – Transport Stream syntax diagram

F.0.2 PES packet

See Figure F.2.

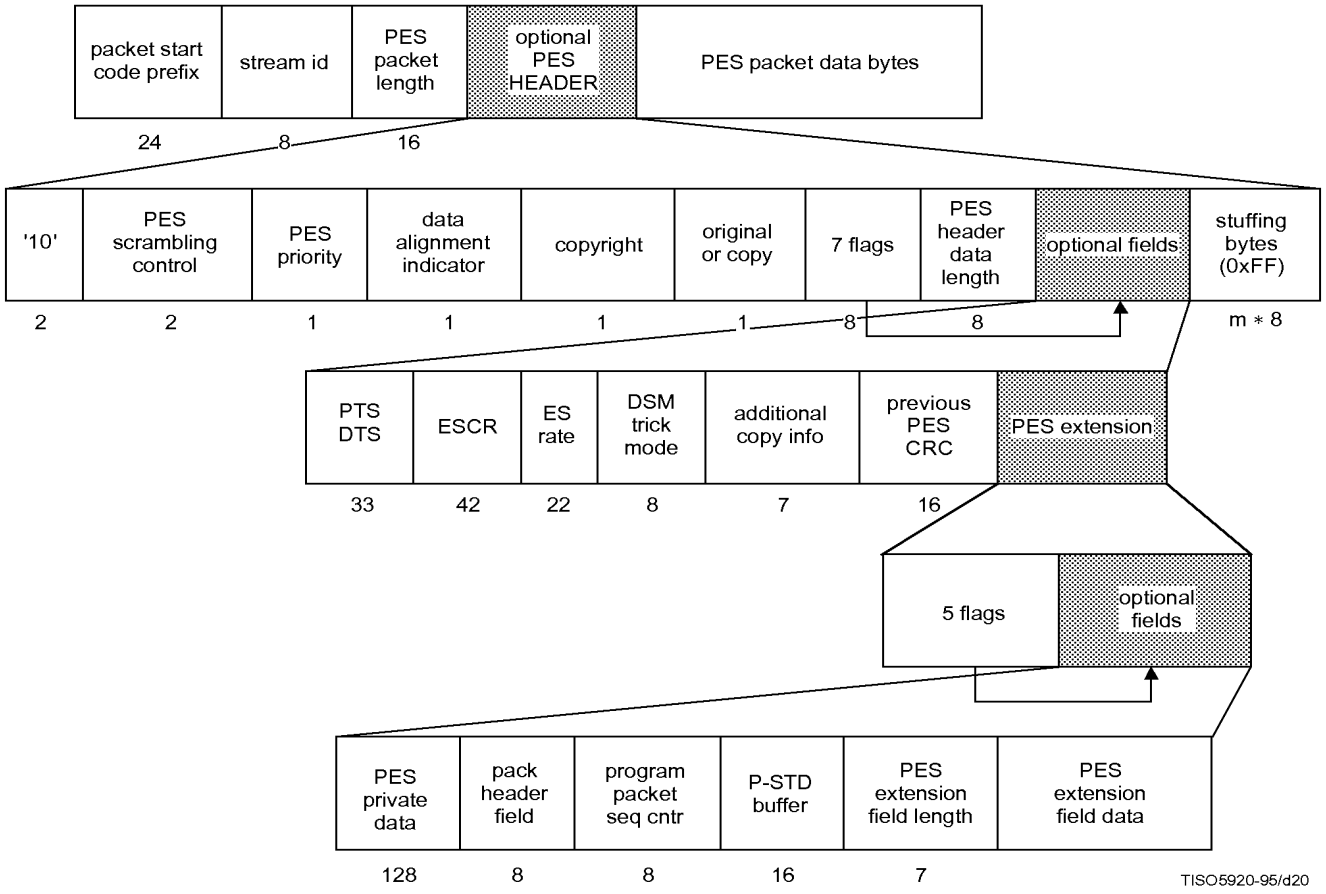


Figure F.2 – PES packet syntax diagram

F.0.3 Program Association Section

See Figure F.3.

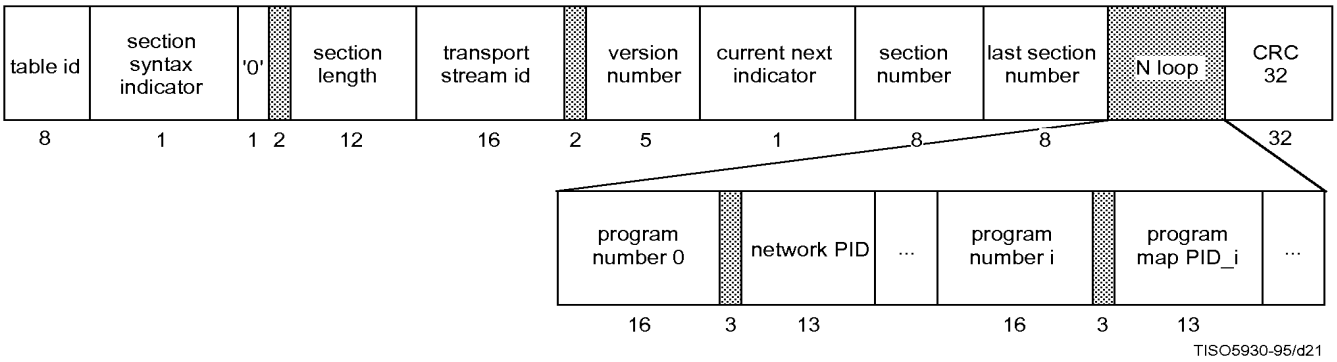


Figure F.3 – Program association section diagram

F.0.4 CA section

See Figure F.4.

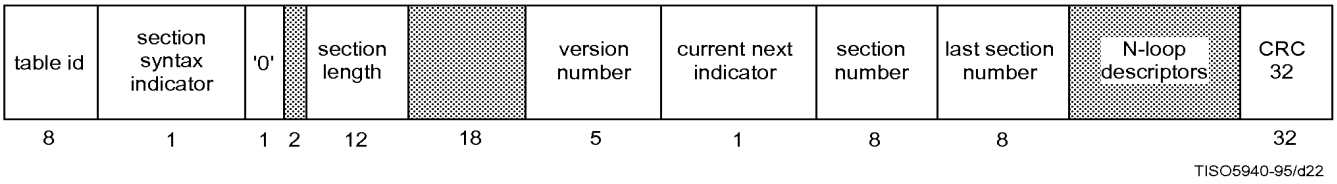


Figure F.4 – Conditional access section diagram

F.0.5 TS program map section

See Figure F.5.

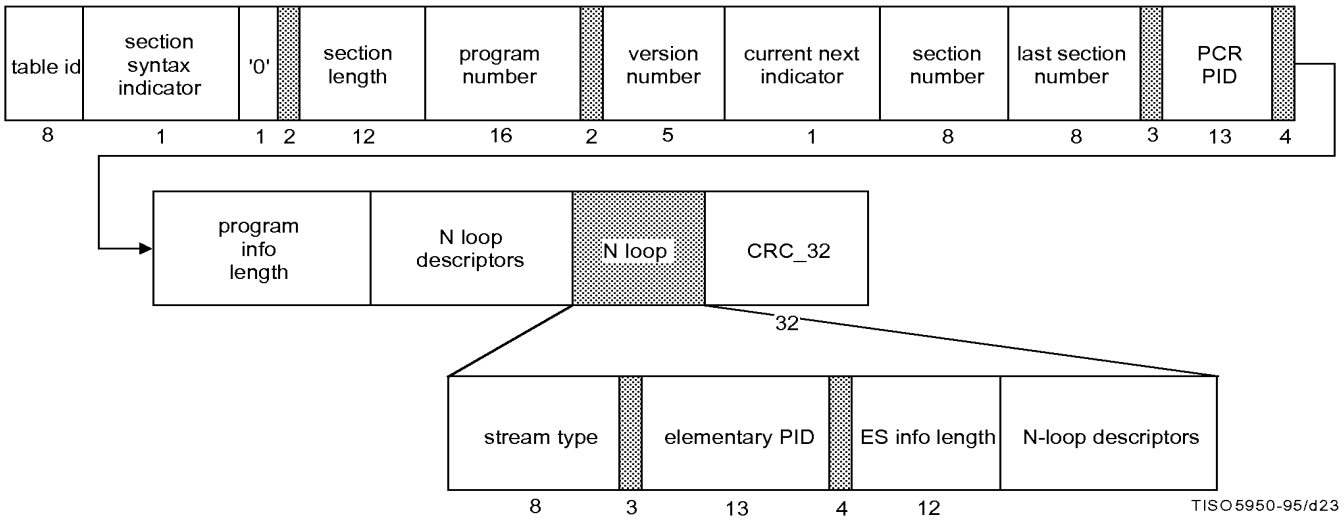


Figure F.5 – TS program map section diagram

F.0.6 Private section

See Figure F.6.

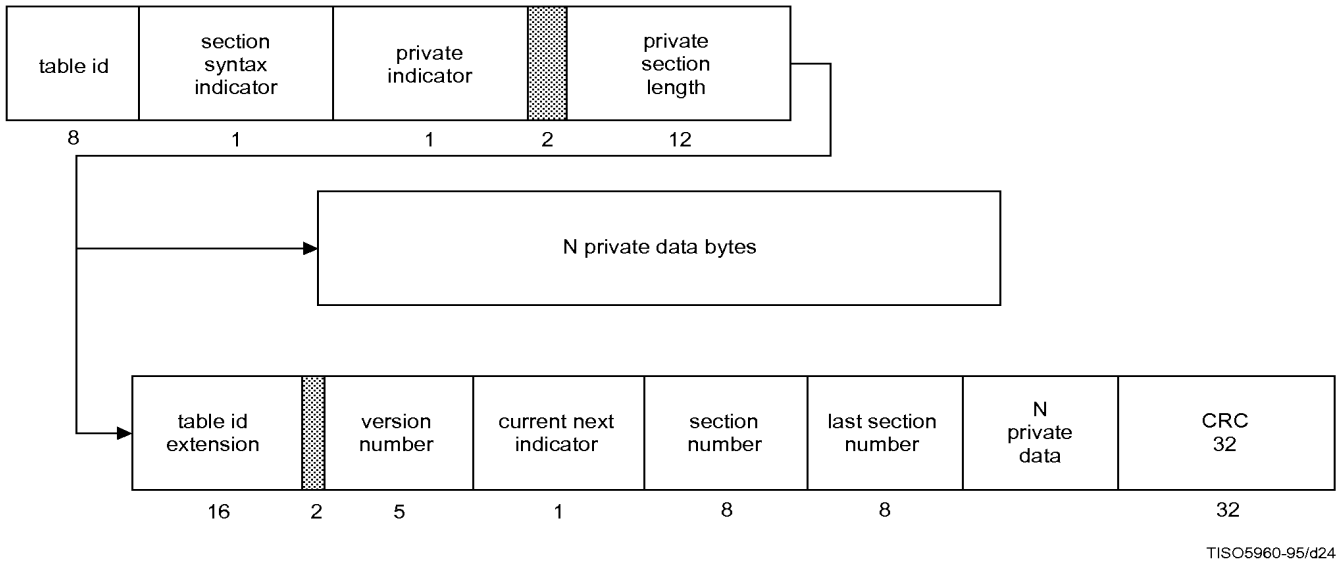


Figure F.6 – Private section diagram

F.0.7 Program Stream

See Figure F.7.

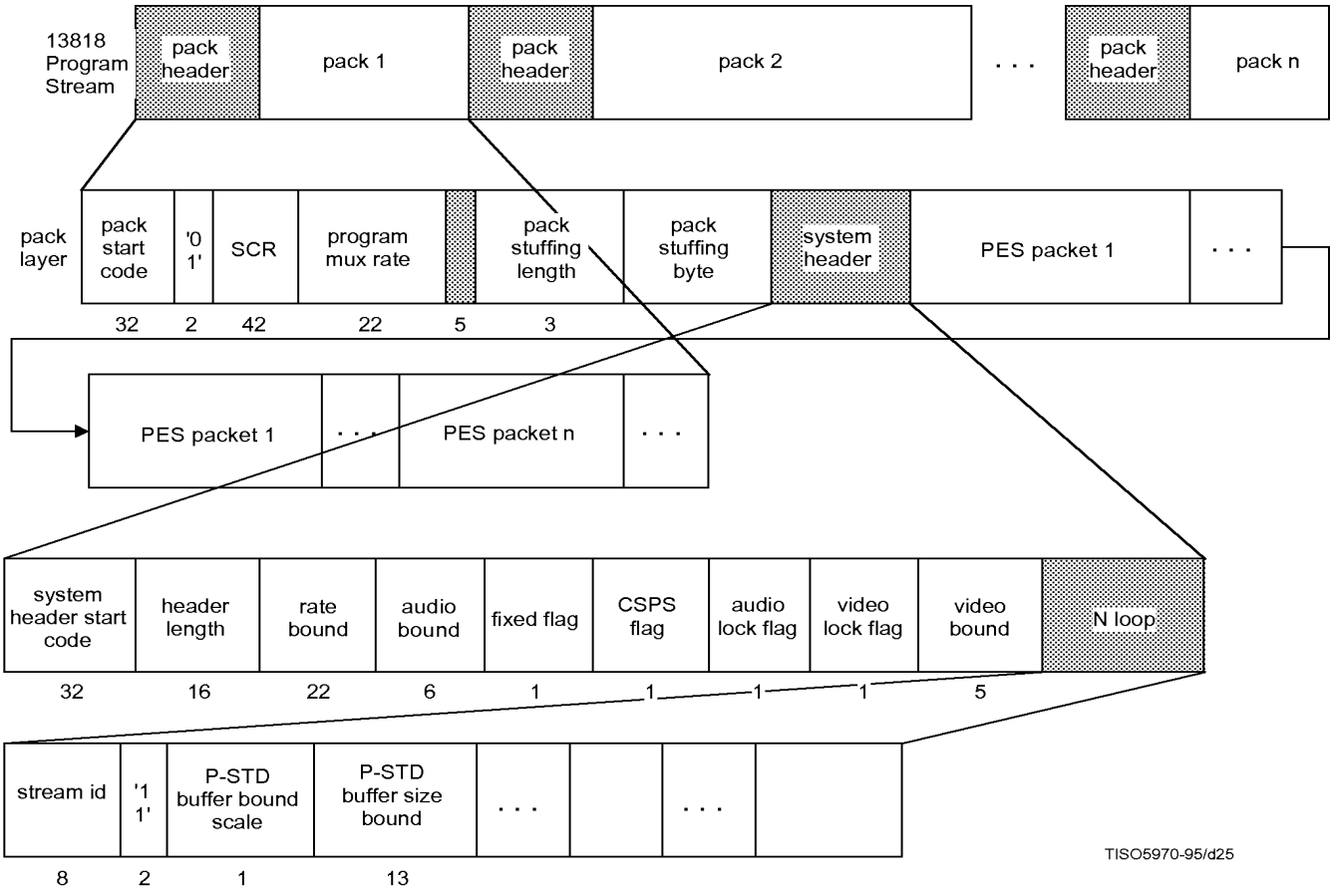


Figure F.7 – Program Stream diagram

F.0.8 Program Stream map

See Figure F.8.

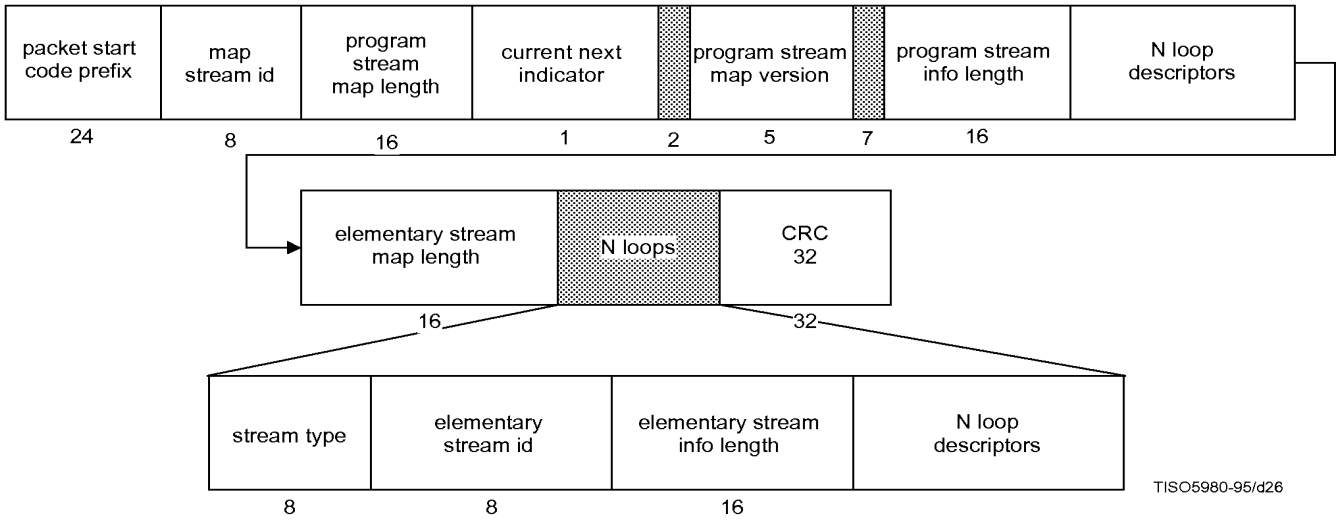


Figure F.8 – Program Stream map diagram

Annex G

General Information

(This annex does not form an integral part of this Recommendation | International Standard)

G.0 General Information

G.0.1 Sync Byte Emulation

In the choice of PID values it is recommended that the periodic emulation of sync bytes be avoided. Such emulation may potentially occur within the PID field or as a combination of the PID field and adjacent flag settings. It is recommended that emulation of the sync byte be permitted to occur in the same position of the packet header for a maximum of 4-consecutive transport packets.

G.0.2 Skipped picture status and decoding process

Assume that the sequence being displayed contains only I- and P-frames. Denote the next picture to be decoded by picture_next, and the picture currently being displayed by picture_current. Because of the fact that the video encoder may skip pictures, it is possible that not all of the bits of picture_next are present in the STD buffers EB_n or B_n when the time arrives to remove those bits for instantaneous decoding and display. When this case arises, no bits are removed from the buffer and picture_current is displayed again. When the next picture display time arrives, if the remainder of the bits corresponding to picture_next are now in buffer EB_n or B_n, all the bits of picture_next are removed and picture_next is displayed. If all the bits of picture_next are not in the buffer EB_n or B_n, the above process of re-displaying picture_current is repeated. This process is repeated until picture_next can be displayed. Note that if a PTS preceded picture_next in the bitstream, it will be incorrect by some multiple of the picture display interval, which itself may depend on some parameters, and must be ignored.

Whenever the skipped picture situation described above occurs, the encoder is required to insert a PTS before the picture to be decoded after picture_next. This allows the decoder to immediately verify that it has correctly displayed the received picture sequence.

G.0.3 Selection of PID Values

Applications are encouraged to use low numbered PID values (avoiding reserved values as specified in Table 2-4) and group values together as much as possible.

G.0.4 PES start_code emulation

Three consecutive bytes having the value of a packet_start_code_prefix (0x000001), which when concatenated with a fourth byte, may emulate the four bytes of a PES_packet_header at a unintended place in the stream.

Such, so called, start code emulation is not possible in video elementary streams. It is possible in audio and data elementary streams. It is also possible at the boundary of a PES_packet_header and a PES_packet payload, even if the PES_packet payload is video.

Annex H

Private Data

(This annex does not form an integral part of this Recommendation | International Standard)

H.0 Private Data

Private data is any user data which is not coded according to a standard specified by ITU-T | ISO/IEC and referred to in this Specification. The contents of this data is not and shall not be specified within this Recommendation | International Standard in the future. The STD defined in this Specification does not cover private data other than the demultiplex process. A private party may define each STD for private streams.

Private data may be carried in the following locations within the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 syntax.

1) *Transport Stream packet Table 2-2*

   The data bytes of the transport_packet() syntax may contain private data. Private data carried in this format is referred to as user private within the stream_type Table 2-29. It is permitted for Transport Stream packets containing private data to also include adaptation_field()s.

2) *Transport Stream Adaptation Field Table 2-6*

   The presence of any optional private_data_bytes in the adaptation_field() is signalled by the transport_private_data_flag. The number of the private_data_bytes is inherently restricted by the semantic of the adaptation_field_length field, where the value of the adaptation_field_length shall not exceed 183 bytes.

3) *PES packet Table 2-17*

   There are two possibilities for carrying private data within PES packets. The first possibility is within the PES_packet_header, within the optional 16 bytes of PES_private_data. The presence of this field is signalled by the PES_private_data_flag. The presence of the PES_private_data_flag is signalled by the PES_extension_flag. If present, these bytes, when considered with the adjacent fields, shall not emulate the packet_start_code_prefix.

   The second possibility is within the PES_packet_data_byte field. This may be referred to as private data within PES packets under the stream_type Table 2-29. This category of private data can be split in two: private_stream_1 refers to private data within PES packets which follow the PES_packet() syntax such that all fields up to and including, but not limited to, PES_header_data_length are present. private_stream_2 refers to private data within PES packets where only the first three fields shall be present followed by the PES_packet_data_bytes containing private data.

   Note that PES packets exist within both Program Streams and Transport Streams therefore private_stream_1 and private_stream_2 exist within both Program Streams and Transport Streams.

4) *Descriptors*

   Descriptors exist within Program Streams and Transport Streams. A range of private descriptors may be defined by the user. These descriptors shall commence with descriptor_tag and descriptor_length fields. For private descriptors, the value of descriptor_tag may take the values 64-55 as identified in Table 2-39. These descriptors may be placed within a program_stream_map() Table 2-29, a CA_section() Table 2-27, a TS_program_map_section(), Table 2-28 and in any private section(), Table 2-30.

   Specifically private_data_bytes also appear in the CA_descriptor().

5) *Private Section*

   The private_section Table 2-30 provides a further means to carry private data also in two forms. This type of elementary stream may be identified under stream_type Table 2-29 as private_data in PSI sections. One type of private_section() includes only the first five defined fields, and is followed by private data. For this structure the section_syntax_indicator shall be set to a value of '0'. For the other type, the section_syntax_indicator shall be set to a value of '1' and the full syntax up to and including last_section_number shall be present, followed by private_data_bytes and ending with the CRC_32.

Annex I

Systems conformance and real-time interface

(This annex does not form an integral part of this Recommendation | International Standard)

I.0 Systems conformance and real-time interface

Conformance for ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Streams and Transport Streams is specified in terms of the normative specifications in this Recommendation | International Standard. These specifications include, among other requirements, a System Target Decoder (T-STD and P-STD) which specifies the behaviour of an idealized decoder when the stream is the input to such a decoder. This model, and the associated verification, do not include information concerning the real-time delivery performance of the stream, except for the accuracy of the system clock frequency which is represented by the Transport Stream and the Program Stream. All Transport Streams and Program Streams must comply with this Recommendation | International Standard.

In addition, there is a real-time interface specification for input of Transport Streams and Program Streams to a decoder. This Recommendation | International Standard allows standardization of the interface between MPEG decoders and adapters to networks, channels, or storage media. The timing effects of channels, and the inability of practical adapters to eliminate completely these effects, causes deviations from the idealized byte delivery schedule to occur. While it is not necessary for all MPEG decoders to implement this interface, implementations which include the interface shall adhere to the specifications. This Recommendation | International Standard covers the real-time delivery behaviour of Transport Streams and Program streams to decoders, such that the coded data buffers in decoders are guaranteed not to overflow nor underflow, and decoders are guaranteed to be able to perform clock recovery with the performance required by their applications.

The MPEG real-time interface specifies the maximum allowable amount of deviation from the idealized byte delivery schedule which is indicated by the Program Clock Reference (PCR) and System Clock Reference (SCR) fields encoded in the stream.

Annex J

Interfacing Jitter-Inducing Networks to MPEG-2 Decoders

(This annex does not form an integral part of this Recommendation | International Standard)

J.0 Introduction

In this annex the expression system stream will be used to refer to both ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Transport Streams and ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Streams. When the term STD is used, it is understood to mean the P-STD (Program System Target Decoder) for Program Streams and the T-STD (Transport System Target Decoder) for Transport Streams.

The intended byte delivery schedule of a system stream can be deduced by analyzing the stream. A system stream is compliant if it can be decoded by the STD, which is a mathematical model of an idealized decoder. If a compliant system stream is transmitted over a jitter-inducing network, the true byte delivery schedule may differ significantly from the intended byte delivery schedule. In such cases it may not be possible to decode the system stream on such an idealized decoder, because jitter may cause buffer overflows or underflows and may make it difficult to recover the time base. An important example of such a jitter-inducing network is ATM.

The purpose of this annex is to provide guidance and insight to entities concerned with sending system streams over jitter-inducing networks. Network specific compliance models for transporting system streams are likely to be developed for several types of networks, including ATM. The STD plus a real-time interface definition can play an integral role in defining such models. A framework for developing network compliance models is presented in J.2.

Three examples of network encoding to enable the building of jitter-smoothing network adapters are discussed in J.3. In the first example a constant bitrate system stream is assumed and a FIFO is used for jitter smoothing. In the second example the network adaptation layer includes timestamps to facilitate jitter smoothing. In the final example a common network clock is assumed to be available end-to-end, and is exploited to achieve jitter smoothing.

J.4 presents two examples of decoder implementations in which network-induced jitter can be accommodated. In the first example, a jitter-smoothing network adapter is inserted between a network's output and an MPEG-2 decoder. The MPEG-2 decoder is assumed to conform to a real-time MPEG-2 interface specification. This interface requires an MPEG-2 decoder with more jitter tolerance than the idealized decoder of the STD. The network adapter processes the incoming jittered bitstream and outputs a system stream whose true byte delivery schedule conforms to the real-time specification. Example one is discussed in J.4.1. For some applications the network adapter approach will be too costly because it requires two stages of processing. Therefore, in the second example the dejittering and MPEG-2 decoding functions are integrated. The intermediate processing of the jitter-removal device is bypassed, so only a single stage of clock recovery is required. Decoders that perform integrated dejittering and decoding are referred to in this annex as integrated network-specific decoders, or simply integrated decoders. Integrated decoders are discussed in J.4.2.

In order to build either network adapters or integrated decoders a maximum value for the peak-to-peak network jitter must be assumed. In order to promote interoperability, a peak-to-peak jitter bound must be specified for each relevant network type.

J.1 Network compliance models

One way to model the transmission of a system stream across a jitter-inducing network is shown in Figure J.1.

The system stream is input to a network-specific encoding device that converts the system stream into a network specific format. Information to assist in jitter removal at the network output may be part of this format. The network decoder comprises a network-specific decoder and an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 decoder. The ITU-T Rec. H.222.0 | ISO/IEC 13818-1 decoder is assumed to conform to a real-time interface specification, and could have the same architecture as the STD with appropriate buffers made larger to provide more jitter tolerance. The network-specific decoder removes the non- ITU-T Rec. H.222.0 | ISO/IEC 13818-1 data added by the network-specific encoder and dejitters the network's output. The output of the network-specific decoder is a system stream that conforms to the real-time specification.

A network target decoder (NTD) can be defined based on the above architecture. A compliant network bitstream would be one that was able to be decoded by the NTD. A network decoder would be compliant provided it could decode any network bitstream able to be decoded by the NTD. A real network decoder might or might not have the architecture of the NTD.

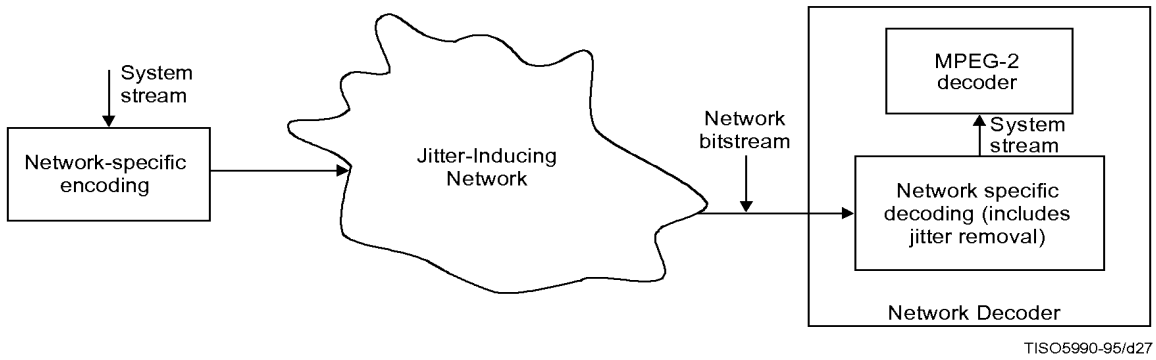


Figure J.1 – Sending system streams over a jitter-inducing network

J.2 Network specification for jitter smoothing

In the case of constant bit rate system streams, jitter smoothing can be accomplished with a FIFO. Additional data that provides specific support for dejittering is not required in the network adaptation layer. After the bytes added by the network encoding are removed, the system stream data is placed in a FIFO. A PLL keeps the buffer approximately half full by adjusting the output rate in response to changes in buffer fullness. In this example the amount of jitter-smoothing achieved will depend on the size of the FIFO and the characteristics of the PLL.

Figure J.2 illustrates a second way to accomplish jitter smoothing. In this example timestamp support from a network adaptation layer is assumed. Using this technique, both constant bit rate and variable bit rate system streams can be dejittered.

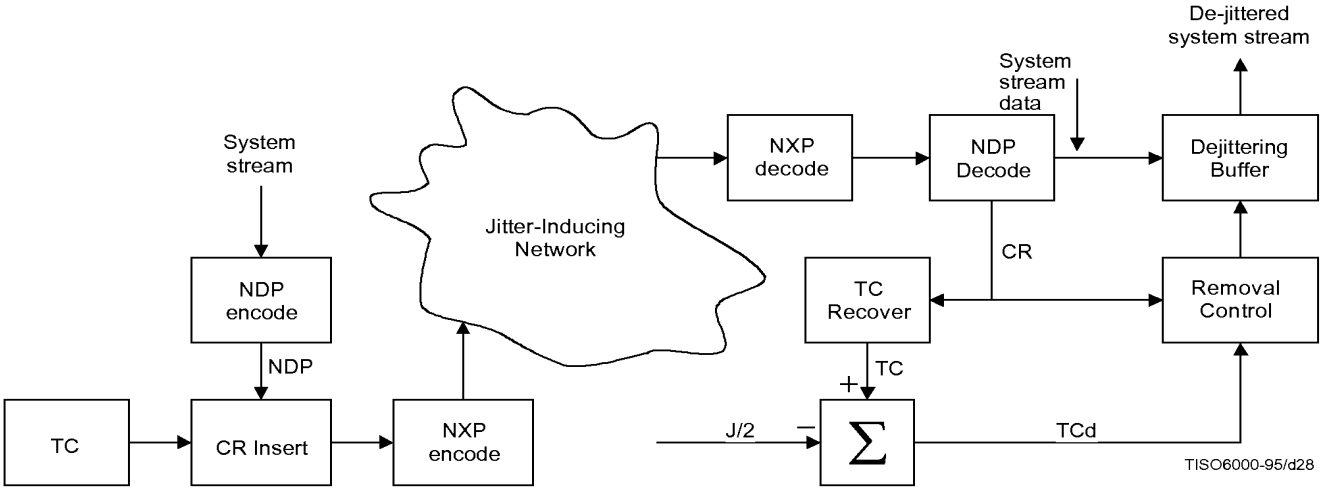


Figure J.2 – Jitter-smoothing using network-layer timestamps

Assume the network adapter is designed to compensate for a peak-to-peak jitter of J seconds. The intended byte delivery schedule is reconstructed using Clock Reference (CRs) samples taken from a Time Clock (TC). The CRs and the TC are analogous to PCRs and the STC. The Network Data Packet (NDP) encode converts each system stream packet into a Network Data Packet (NDP). The network data packets contain a field for carrying CR values, and the current value of the TC is inserted into this field as the NDP leaves the NDP encoder. The Network Transport Packetization (NXP) function encapsulates the NDPs into network transport packets. After transmission across the network, the CRs are extracted by the NDP decoder as the NDPs enter the NDP decoder. The CRs are used to reconstruct the TC, for example by using a PLL. The first MPEG-2 packet is removed from the dejittering buffer when the delayed TC (TCd) is equal to the first MPEG-2 packet's CR. Subsequent MPEG-2 packets are removed when their CR values equal the value of the TCd.

Ignoring implementation details such as the speed of the TC clock recovery loop and the spectral purity of the TC, the size of the dejittering buffer depends only on the maximum peak-to-peak jitter to be smoothed and the largest transport rate that occurs in the system stream. The dejittering buffer size, B_{dj}, is given by

$$B_{dj} = JR_{max}$$

where R_{max} is the maximum data rate of the system stream in bits per second. When packets traversing the network experience the nominal delay, the buffer is half full. When they experience a delay of J/2 seconds, the buffer is empty, and when they experience a delay (advance) of –J/2 seconds the buffer is full.

As a final example, in some cases a common network clock will be available end-to-end, and it may be feasible to lock the system clock frequency to the common clock. The network adapter can smooth jitter with a FIFO. The adapter uses PCRs or SCRs to reconstruct the original byte delivery schedule.

J.3 Example decoder implementations

J.3.1 Network adapter followed by an MPEG-2 decoder

In this implementation a network adapter conforming to the network compliance specification is connected to an MPEG-2 decoder conforming to the real-time interface specification.

J.3.2 Integrated decoder

The example presented in J.4.1 requires two stages of processing. The first stage is necessary to dejitter the network's output. The second stage, recovering the STC by processing PCRs or SCRs, is required for STD decoding. The example presented in this subclause is a decoder that integrates the dejittering and decoding functions in a single system. The STC clock is recovered directly using the jittered PCR or SCR values. For presenting this example, an MPEG-2 transport stream will be assumed.

Figure J.3 illustrates the operation of the integrated decoder. The stream of network packets input to the decoder is assumed to be the same as the one shown in Figure J.2.

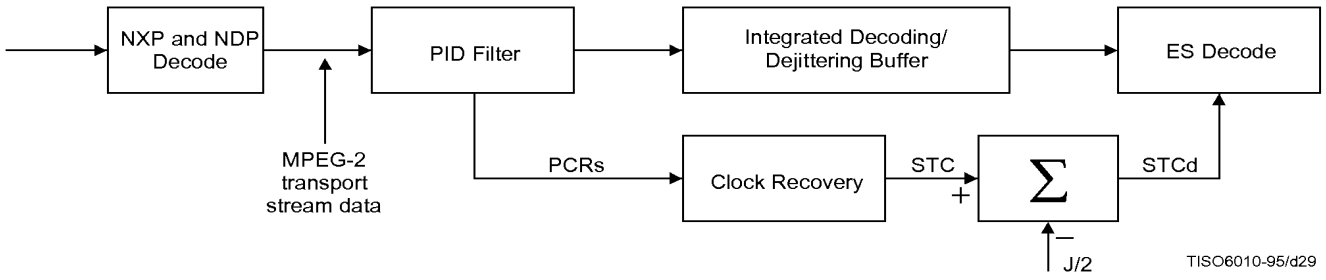


Figure J.3 – Integrated dejittering and MPEG-2 decoding

The incoming network packets are reassembled into MPEG-2 transport stream data by the NXP and NDP decode functions. The jittered ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Transport Stream packets are then filtered to extract packets with the desired PID. For the case illustrated, the PID being decoded is also carrying the PCRs. The PCR values are sent to a PLL to recover the STC. Entire packets for the selected PID are placed in the integrated buffer. A positive value of J/2 s is subtracted from the STC to obtain the delayed STC, STCd. Again, J is the peak-to-peak jitter the network-savvy decoder can accommodate. The delay is introduced to guarantee that all the data required for an access unit has arrived in the buffer when the PTS/DTS of the access unit equals the current value of the STCd.

Ignoring implementation details such as the speed of the STC clock recovery loop and the spectral purity of the STC:

$$\begin{aligned} B_{size} &= B_{dec} + B_{mux} + B_{OH} + 512 + B_j \\ &= B_n + 512 + B_j \end{aligned}$$

where $B_j = R_{max} J$ and R_{max} the maximum rate at which data is input to the PID filter. Depending on the implementation, the integrated memory could be broken into two components as in the transport STD.

Annex K

Splicing Transport Streams

(This annex does not form an integral part of this Recommendation | International Standard)

K.0 Introduction

For the purposes of this annex, the term 'splicing' refers to the concatenation performed on the Transport level of two different elementary streams, the resulting Transport Stream conforming totally to this Recommendation | International Standard. The two elementary streams may have been generated at different locations and/or at different times, and were not necessarily intended to be spliced together when they were generated. In the following we will call the 'old' stream a continuous elementary stream (video or audio), which has been superseded by another stream (the 'new' one) from a certain point on. This point is called the splice. It is the boundary between data belonging to the 'old' stream and data belonging to the 'new' one.

A splice can be seamless or non-seamless:

- A seamless splice is a splice inducing no decoding discontinuity (refer to 2.7.6). This means that the decoding time of the first access unit of the 'new' stream is consistent with respect to the decoding time of the access unit of the 'old' stream preceding the splice, i.e. it is equal to the one that the next access unit would have had if the 'old' stream had continued. In the following, we will call this decoding time the 'seamless decoding time'.
- A non-seamless splice is a splice which results in a decoding discontinuity, i.e. the decoding time of the first access unit of the 'new' stream is greater than the seamless decoding time.

  NOTE – A decoding time lower than the seamless decoding time is forbidden.

Splicing is allowed to be performed at any transport stream packet boundary, since the resulting stream is legal. But in a general case, if nothing is known about the location of PES packet starts and access unit starts, this constraint imposes that not only the Transport layer is parsed, but also the PES layer and the Elementary Stream layer, and may in some cases, make some processing on the payload of Transport Stream packets necessary. If such complex operations are wished to be avoided, splicing should be performed at locations where the Transport Stream has favourable properties, these properties being indicated by the presence of a splicing point.

The presence of a splicing point is indicated by the splice_flag and splice_countdown fields (refer to 2.4.3.4 for the semantics of these fields). In the following, the Transport Stream packet in which the splice_countdown field value reaches zero will be called 'splicing packet'. The splicing point is located immediately after the last byte of the splicing packet.

K.1 The different types of splicing point

A splicing point can be either an ordinary splicing point or a seamless splicing point.

K.1.1 Ordinary splicing points

If the seamless_splice_flag field is not present, or if its value is zero, the splicing point is ordinary. The presence of an ordinary splicing point only signals alignment properties of the Elementary Stream: the splicing packet ends on the last byte of an Access Unit, and the payload of the next Transport Stream packet of the same PID will start with the header of a PES packet, the payload of which will start with an Elementary Stream Access Point (or with a sequence_end_code() immediately followed by an Elementary Stream Access Point, in the case of video). These properties allow 'Cut and Paste' operations to be performed easily on the Transport level, while respecting syntactical constraints and ensuring bit stream consistency. However, it does not provide any information concerning timing or buffer properties. As a consequence, with such splicing points, seamless splicing can only be done with the help of private arrangements, or by analyzing the payload of the Transport Stream Packets and tracking buffer status and timestamp values.

K.1.2 Seamless splicing points

If the seamless_splice_flag field is present and its value is one, information is given by the splicing point, indicating some properties of the 'old' stream. This information is not aimed at decoders. Its primary goal is to facilitate seamless splicing. Such a splicing point is called a seamless splicing point. The available information is:

- The seamless decoding time, which is encoded as a DTS value in the DTS_next_AU field. This DTS value is expressed in the time base which is valid in the splicing packet.
- In the case of a video elementary stream, the constraints that have been applied to the 'old' stream when it was generated, aiming at facilitating seamless splicing. These conditions are given by the value of the splice_type field, in the table corresponding to the profile and level of the video stream.

Note that a seamless splicing point can be used as an ordinary splicing point, by discarding this additional information. This information may also be used if judged helpful to perform non-seamless splicing, or for purposes other than splicing.

K.2 Decoder behaviour on splices

K.2.1 On non-seamless splices

As described above, a non-seamless splice is a splice which results in a decoding discontinuity.

It shall be noted that with such a splice, the constraints related to the decoding discontinuity (see 2.7.6) shall be fulfilled. In particular:

- a PTS shall be encoded for the first access unit of the 'new' stream (except during trick mode operation or when low_delay = '1');
- the decoding time derived from this PTS (or from the associated DTS) shall not be earlier than the seamless decoding time;
- in the case of a video elementary stream, if the splicing packet does not end on a sequence_end_code(), the 'new' stream shall begin with a sequence_end_code() immediately followed by a sequence_header().

In theory, since they introduce decoding discontinuities, such splices result in a non-continuous presentation of presentation units (i.e. a variable length dead time between the display of two consecutive pictures, or between two consecutive audio frames). In practice, the result will depend on how the decoder is implemented, especially in video. With some video decoders, the freezing of one or more pictures may be the preferred solution. See Part 4 of ISO/IEC 13818.

K.2.2 On seamless splices

The aim of having no decoding discontinuity is to allow having no presentation discontinuity. In the case of audio, this can always be ensured. But it has to be noted that in the case of video, presentation continuity is in theory not possible in cases 1) and 2) below:

1) The 'old' stream ends on the end of a low-delay sequence, and the 'new' stream begins with the start of a non-low-delay sequence.
2) The 'new' stream ends on the end of a non-low-delay sequence, and the 'new' stream begins with the start of a low-delay sequence.

The effects induced by such situations is implementation dependent. For instance, in case 1, a picture may have to be presented during two frame periods, and in case 2, a picture may have to be skipped. However, it is technically possible that some implementations support such situations without any undesirable effect.

In addition, referring to 6.1.1.6 of ITU-T Rec. H.262 | ISO/IEC 13818-2, a sequence_end_code() shall be present before the first sequence_header() of the 'new' stream, if at least one sequence parameter (i.e. a parameter defined in the sequence header or in a sequence header extension) has a different value in both streams, with the only exception of those defining the quantization matrix. As an example, if the bit rate field has not the same value in the 'new' stream as in the 'old' one, a sequence_end_code() shall be present. Thus, if the splicing packet does not end on a sequence_end_code, the 'new' stream shall begin with a sequence_end_code followed by a sequence_header.

According to the previous paragraph, a sequence_end_code will be mandatory in most splices, even seamless ones. It has to be noted that ITU-T Rec. H.262 | ISO/IEC 13818-2 specifies the decoding process of video sequences (i.e. data comprised between a sequence_header() and a sequence_end_code()), and nothing is specified about how to handle a sequence change. Thus, for the behaviour of the decoders when such splices are encountered, refer to Part 4 of ISO/IEC 13818.

K.2.3 Buffer Overflow

Even if both elementary streams obey the T-STD model before being spliced, it is not necessarily ensured that the STD buffers do not overflow with the spliced stream in the time interval during which bits of both streams are in these buffers.

In the case of constant bit rate video, if no particular conditions have been applied to the 'old' stream, and if no particular precautions have been taken during splicing, this overflow is possible in the case where the video bit rate of the 'new' stream is greater than the video bit rate of the 'old' one. Indeed, it is certainly true that the buffers MB_n and EB_n of the T-STD do not overflow if bits are delivered to the T-STD at the 'old' rate. But if the delivery rate is switched to a higher value at the input of TB_n before 'old' bits are completely removed from the T-STD, the fullness of the STD buffers will become higher than if the 'old' stream had continued without splicing, and may cause overflow of EB_n and/or MB_n. In the case of variable bit rate video, the same problem can occur if the delivery rate of the 'new' stream is higher than the one for which provision was made during the creation of the 'old' stream. Such a situation is forbidden.

However, it is possible for the encoder generating the 'old' stream to add conditions in the VBV buffer management in the neighborhood of splicing points, so that provision is made for any 'new' video bit rate lower than a chosen value. For instance, in the case of a seamless splicing point, such additional conditions can be indicated by a 'splice_type' value to which entries correspond in Table 2-7 through Table 2-16 for 'splice_decoding_delay' and 'max_splice_rate'. In that case, if the video bit rate of the 'new' stream is lower than 'max_splice_rate', it is ensured that the spliced stream will not lead to overflow during the time interval during which bits of both streams are in the T-STD buffer.

In the case where no such constraints have been applied, this problem can be avoided by introducing a dead time in the delivery of bits between the 'old' stream and the 'new' one, in order to let the T-STD buffers get sufficiently empty before the bits of the 'new' stream are delivered. If we call t_{in} the time at which the last byte of the last access unit of the 'old' stream enters the STD, and t_{out} the time at which it exits the STD, it is sufficient to ensure that no more bits enter the T-TD the time interval $[t_{in}, t_{out}]$ with the spliced stream than if the 'old' stream had continued without splicing. As an example, in the case where the 'old' stream has a constant bit rate R_{old}, and the 'new' one a constant bit rate R_{new}, it is sufficient to introduce a dead time T_d satisfying the following relations to avoid this risk of overflow:

$$T_d \geq 0 \text{ and } T_d \geq (t_{out} - t_{in}) \times (1 - R_{old}/R_{new})$$

Annex L

Registration procedure (see 2.9)

(This annex does not form an integral part of this Recommendation | International Standard)

L.1 Procedure for the request of a Registered Identifier (RID)

Requesters of a RID shall apply to the Registration Authority. Registration forms shall be available from the Registration Authority. Information which the requester shall provide is given in L.3. Companies and organizations are eligible to apply.

L.2 Responsibilities of the Registration Authority

The primary responsibilities of the Registration Authority administrating the registration of copyright_identifiers is outlined in this subclause; certain other responsibilities may be found in the JTC 1 Directives. The Registration Authority shall:

a) implement a registration procedure for application for a unique RID in accordance with Annex H/JTC 1 Directives;

b) receive and process the applications for allocation of the work type code identifier from Copyright Registration Authority;

c) ascertain which applications received are in accordance with this registration procedure, and to inform the requester within 30 days of receipt of the application of their assigned RID;

d) inform application providers whose request is denied in writing within 30 days of receipt of the application, and also inform the requesting party of the appeals process;

e) maintain an accurate register of the allocated RID. Revisions to the contact information and technical specifications shall be accepted and maintained by the Registration Authority;

f) make the contents of this register available upon request to any interested party;

g) maintain a database of RID request forms, granted and denied. Parties seeking technical information on the format of private data which has a copyright_identifier shall have access to such information which is part of the database maintained by the Registration Authority;

h) report its activities to JTC 1, the ITTF and the JTC 1/SC 29 Secretariat, or their respective assignees, annually on a schedule mutually agreed upon.

L.2.1 Contact information of the Registration Authority

Organization Name:

Address:

Telephone:

Fax:

L.3 Responsibilities of parties requesting an RID

The party requesting an RID for the purpose of copyright identification shall:

a) apply using the form and procedures supplied by the Registration Authority;

b) provide contact information describing how a complete description of the copyright organization can be obtained on a non-discriminatory basis;

c) include technical details of the syntax and semantics of the data format used to describe the audiovisual works or other copyrighted works within the **additional_copyright_info** field. Once registered, the syntax used for the additional copyright information shall not change;

d) agree to institute the intended use of the granted copyright_identifier within a reasonable time-frame;

e) maintain a permanent record of the application form and the notification received from the Registration Authority of each granted copyright_identifier.

L.4 Appeal procedure for denied applications

The Registration Management Group is formed to have jurisdiction over appeals relating to a denied request for an RID. The RMG shall have a membership who are nominated by P and L members of the ISO technical body responsible for this Recommendation | International Standard. It shall have a convenor and secretariat nominated from its members. The Registration Authority is entitled to nominate one non-voting observing member.

The responsibilities of the RMG shall be:

a) to review and act on all appeals within a reasonable time-frame;

b) to inform, in writing, organizations which make an appeal for reconsideration of its petition of the RMG's disposition of the matter;

c) to review the annual report of the Registration Authority summary of activities;

d) to supply ISO member bodies with information concerning the scope of operation of the Registration Authority.

Annex M

Registration application form (see 2.9)

(This annex does not form an integral part of this Recommendation | International Standard)

M.1 Contact information of organization requesting a Registered Identifier (RID)

Organization Name:

Address:

Telephone:

Fax:

email:

M.2 Statement of an intention to apply the assigned RID

RID application domain: using guidelines to be provided by the Registration Authority.

M.3 Date of intended implementation of the RID

M.4 Authorized representative

Name:

Title:

Address:

Signature: _____

M.5 For official use only of the Registration Authority

Registration rejected: _____ Reason for rejection of the application: Registration granted: _____ Registration value: _____

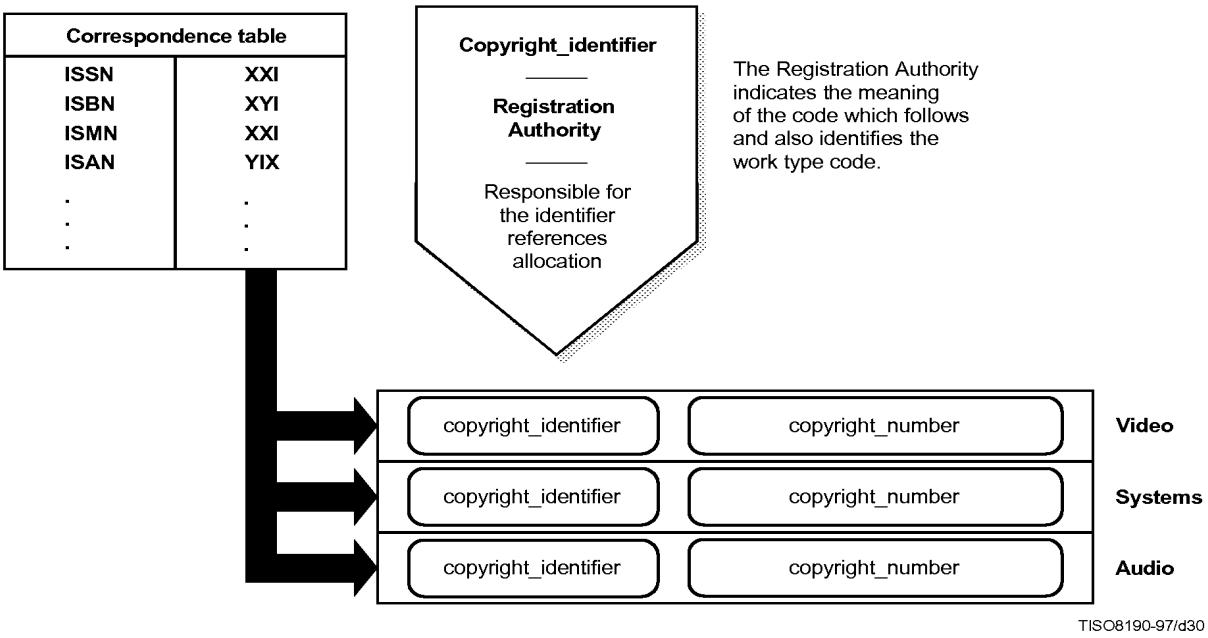
Attachment 1 – Attachment of technical details of the registered data format.

Attachment 2 – Attachment of notification of appeal procedure for rejected applications.

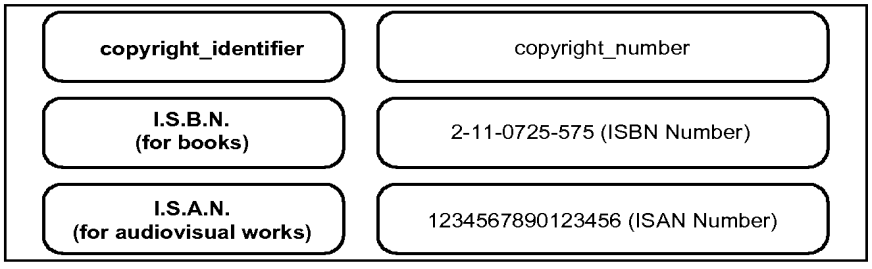
Annex N

(This annex does not form an integral part of this Recommendation | International Standard)

Registration Authority
Diagram of administration structure (see 2.9)



Examples



All the copyright_identifiers are registered by the Registration Authority, uniquely for copyright_numbers standardized by ISO. Each organization which allocates copyright_numbers, requests a specific copyright_identifier from the Registration Authority, e.g. Staatsbibliothek Preussischer Kulturbesitz, designated by ISO to manage I.S.B.N., asks for a specific copyright_identifier from the R.A. for book numbering.

Annex O

Registration procedure (see 2.10)

(This annex does not form an integral part of this Recommendation | International Standard)

O.1 Procedure for the request of an RID

Requesters of an RID shall apply to the Registration Authority. Registration forms shall be available from the Registration Authority. The requester shall provide the information specified in O.4. Companies and organizations are eligible to apply.

O.2 Responsibilities of the Registration Authority

The primary responsibilities of the Registration Authority administrating the registration of private data format_identifiers is outlined in this annex; certain other responsibilities may be found in the JTC 1 Directives. The Registration Authority shall:

a) implement a registration procedure for application for a unique RID in accordance with the JTC 1 Directives;

b) receive and process the applications for allocation of an identifier from application providers;

c) ascertain which applications received are in accordance with this registration procedure, and to inform the requester within 30 days of receipt of the application of their assigned RID;

d) inform application providers whose request is denied in writing within 30 days of receipt of the application, and to consider resubmission of the application in a timely manner;

e) maintain an accurate register of the allocated identifiers. Revisions to format specifications shall be accepted and maintained by the Registration Authority;

f) make the contents of this register available upon request to National Bodies of JTC 1 that are members of ISO or IEC, to liaison organizations of ISO or IEC and to any interested party;

g) maintain a database of RID request forms, granted and denied. Parties seeking technical information on the format of private data which has an RID shall have access to such information which is part of the database maintained by the Registration Authority;

h) report its activities to JTC 1, the ITTF, and the SC 29 Secretariat, or their respective designees, annually;

i) accommodate the use of existing RIDs whenever possible.

O.3 Contact information for the Registration Authority

O.4 Responsibilities of parties requesting an RID

The party requesting a format_identifier shall:

a) apply, using the form and procedures supplied by the Registration Authority;

b) include a description of the purpose of the registered bit stream, and the required technical details as specified in the application form;

c) provide contact information describing how a complete description can be obtained on a non-discriminatory basis;

d) agree to institute the intended use of the granted RID within a reasonable time-frame;

e) to maintain a permanent record of the application form and the notification received from the Registration Authority of a granted RID.

O.5 Appeal procedure for denied applications

The Registration Management Group is formed to have jurisdiction over appeals to denied requests for an RID. The RMG shall have a membership who is nominated by P- and L-members of the ISO technical committee responsible for this Specification. It shall have a convenor and secretariat nominated from its members. The Registration Authority is entitled to nominate one non-voting observing member.

The responsibilities of the RMG shall be:

a) to review and act on all appeals within a reasonable time-frame;

b) to inform, in writing, organizations which make an appeal for reconsideration of its petition of the RMG's disposition of the matter;

c) to review the annual report of the Registration Authorities summary of activities;

d) to supply member bodies of ISO and National Committees of IEC with information concerning the scope of operation of the Registration Authority.

Annex P

Registration application form

(This annex does not form an integral part of this Recommendation | International Standard)

P.1 Contact information of organization requesting an RID

Organization Name:

Address:

Telephone:

Fax:

email:

Telex:

P.2 Request for a specific RID

NOTE – If the system has already been implemented and is in use, fill in this item and also the item P.3 and then skip to P.6; otherwise leave this space blank and skip to P.4.

P.3 Short description of RID that is in use and date system that was implemented

P.4 Statement of an intention to apply the assigned RID

P.5 Date of intended implementation of the RID

P.6 Authorized representative

Name:

Title:

Address:

Signature: _____

P.7 For official use of the Registration Authority

Registration rejected: _____

Reason for rejection of the application:

Registration granted: _____ Registration value: _____

Attachment 1 – Attachment of technical details of the registered data format.

Attachment 2 – Attachment of notification of appeal procedure for rejected applications.

Annex Q

T-STD and P-STD buffer models for ISO/IEC 13818-7 ADTS

(This annex does not form an integral part of this Recommendation | International Standard)

Q.1 Introduction

The Transport Stream system target decoder model for audio streams is defined in subclause 2.4.2. In this annex, the buffer model for ISO/IEC 13818-7 ADTS is described.

ISO/IEC 13818-7 ADTS audio streams can be recognized in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 multiplex through the presence of stream_id=0x110yyyyy ('y' = "don't care") and stream_type=0x0F as defined in Tables 2-18 and 2-19.

Q.2 Leak Rate from Transport Buffer

For audio except ISO/IEC 13818-7 ADTS, the leak rate from Transport Buffer is 2 Mbit/s. This rate is, however, lower than the maximum rate of ISO/IEC 13818-7 ADTS. Therefore, the leak rate for ISO/IEC 13818-7 ADTS streams is set to a different value from ISO/IEC 11172-3 and ISO/IEC 13818-3 audio streams.

ISO/IEC 13818-7 ADTS elementary stream consists of one or more channels. The maximum rate of each channel is 576 kbit/s where the sampling frequency is 96 kHz. Therefore, the leak rate for ISO/IEC 13818-7 ADTS is calculated as following equation.

$$Rx_n = 1.2 \times R_{max} \times N \text{ bits per second}$$

where:

R_{max} is a constant 576 kbit/s as defined in subclause 3.2.2 of ISO/IEC 13818-7. It is an upper bound of the bit rate per channel of AAC ADTS stream corresponding to the maximum value of sampling frequency (i.e., Fs = 96 kHz).

And where:

N is the number of audio channels that require their own decoder buffer in this elementary stream (i.e. individual channel streams in a single channel element or channel pair element and independently switched coupling channel elements).

Q.3 Buffer Size

For audio except ISO/IEC 13818-7 ADTS, the main buffer size is 3584 bytes. This size is, however, smaller than the maximum decoder input buffer size of ISO/IEC 13818-7 ADTS. Therefore, the main buffer size for the ISO/IEC 13818-7 ADTS stream is set to a different value from ISO/IEC 11172-3 and ISO/IEC 13818-3 audio streams.

The main buffer size for ISO/IEC 13818-7 ADTS is calculated as following.

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh}$$

Where BS_{oh}, PES packet overhead buffering is defined as:

$$BS_{oh} = 528 \text{ bytes}$$

and BS_{mux}, additional multiplexing buffering is defined as:

$$BS_{mux} = 0.004 \text{ seconds} \times R_{max} \times N$$

and BS_{dec}, access unit buffering is defined as

$$BS_{dec} = 6144 \text{ bits} \times N$$

where

R_{max} is a constant 576 kbit/s as defined in subclause 3.2.2 of ISO/IEC 13818-7. It is an upper bound of the bit rate per channel of AAC ADTS stream corresponding to the maximum value of sampling frequency (i.e., Fs = 96 kHz).

And where

N is the number of audio channels that require their own decoder buffer in this elementary stream (i.e. individual channel streams in a single channel element or channel pair element and independently switched coupling channel elements).

Q.3.1 TBS_n: same as other audio

In term of the smoothing buffer, there is no difference in TB_n between ISO/IEC 13818-7 ADTS and other audio streams. Consequently, it is not necessary to change TBS_n, which is size of TB_n.

Q.3.2 BS_{mux}: different from other audio

BS_{mux}, additional multiplexing buffering, shall be changed to accept up to 4 msec of delay jitter. This is similar to the approach taken for other streams in ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

Q.3.3 BS_{dec}: different from other audio

BS_{dec}, access unit buffering is based on the decoder input buffer size of the elementary stream. As defined in subclause 3.2.2 of ISO/IEC 13818-7, total decoder input buffer size is 6144 bit multiplied by the number of channels which require their each decoder input buffer.

Q.3.4 BS_{oh}: different from other audio

BS_{oh} is corresponding to the PES packet header overhead.

In subclause 2.4.2.6,

> *The delay of any data through the System Target Decoders buffers shall be less than or equal to one second except for still picture video data.*

Besides, in subclause 2.7.4,

> *The program Stream and Transport Stream shall be constructed so that the maximum difference between coded presentation time stamp referring to each elementary video or audio stream is 0.7 seconds.*

BS_{oh} shall be set to the appropriate size corresponding to the PES packet header overhead when AAC stream is packetized with the above rules. The maximum size of PES packet header is 264 bytes. Therefore, BS_{oh} = 528 bytes, i.e. the twice the maximum size of PES packet header, assures that at least two PES packet headers can enter the main buffer regardless of the size of PES packet header. It means that PES packet header with PTS can be inserted at less than 0.7 seconds intervals even when the data of one second will be in the main buffer.

Example: sampling frequency is 48 kHz

The size of PES packet header without any optional fields except PTS is 18 bytes. The number of Access Units of one second is about 47. When the data of one second is in the main buffer (i.e. the worst case), PES packet header overhead can fit the BS_{oh} with packetizing more than or equal to two Access Units into one packet.

number_of_AU = 48 kHz/1024 = 46,875 per second

(number_of_AU/2) × 18[byte] = 421,875 bytes < BS_{oh}

More frequent PES packet headers can fit BS_{oh}, if the delay of any data through the main buffer is shorter than one second.

Q.4 Conclusion

The decoder buffer model should cover the maximum size of buffer, however AAC can handle up to 48 channels and very high bitrate. Therefore the 3 levels of number of channels, 2, 8 and 48, are used to define the leak rate and the main buffer size. In a case of 2, the same leak rate and main buffer size as the conventional values are used to keep the compatibility. In other cases (8 and 48), the proposed formulas are applied.

T-STD leak rate for ISO/IEC 13818-7 ADTS audio,

Number of Channels	Rx_n [bit/s]
1-2	2 000 000
3-8	5 529 600
9-12	8 294 400
13-48	33 177 600

Channels: channels that require their own decoder buffer in this elementary stream (i.e. individual channel streams in a single channel element or channel pair element and independently switched coupling channel elements).

T-STD main buffer size or ISO/IEC 13818-7 ADTS audio,

Number of Channels	BSn [bytes]
1-2	3 584
3-8	8 976
9-12	12 804
13-48	51 216

Channels: channels that require their own decoder buffer in this elementary stream (i.e. individual channel streams in a single channel element or channel pair element and independently switched coupling channel elements).

For Program stream, the above main buffer size should be set in the P-STD_buffer_scale and P-STD_buffer_size as follows.

Number of Channels	P-STD_buffer_scale	P-STD_buffer_size
1-2	0	28
3-8	0	71
9-48	0	401

Annex R

Carriage of ISO/IEC 14496 scenes in ITU-T Rec. H.222.0 | ISO/IEC 13818-1

(This annex does not form an integral part of this Recommendation | International Standard)

R.1 Content access procedure for ISO/IEC 14496 program components within a Program Stream

The following provides a reference receiver acquisition procedure for accessing ISO/IEC 14496 program elements in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Stream. Here, it is assumed that the Program Stream includes a Program Stream Map (conveyed in a PES packet having stream_id equal to 0xBC):

- Acquire the Program Stream Map.
- Identify the IOD descriptor in the first descriptor loop.
- Identify the ES_IDs of the object descriptor, scene description and other streams described within the initial object descriptor.
- Acquire the SL descriptor and FMC descriptor in the second descriptor loop for elementary_stream_ids 0xFA and 0xFB, as applicable.
- Generate from these descriptors a stream map table between ES_IDs and associated elementary_stream_id plus FlexMux channel, if applicable.
- Locate the object descriptor stream using its ES_ID and the stream map table.
- Locate other streams described in the initial object descriptor using their ES_ID and the stream map table.
- Continuously monitor the object descriptor stream and identify ES_IDs of additional streams.
- Locate the additional streams using their ES_ID and the stream map table.

Figure R.1 gives an example of ISO/IEC 14496 content in a Program Stream, consisting of object descriptor stream, scene description stream (BIFS-Command), BIFS-Anim stream and IPMP stream. All ISO/IEC 14496 streams are multiplexed in a single FlexMux stream.

R.2 Content access procedure for ISO/IEC 14496 program components within a Transport Stream

The following provides a reference receiver acquisition procedure for accessing ISO/IEC 14496 program elements in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Transport Stream:

- Acquire the Program Map Table for the desired program.
- Identify the IOD descriptor in the first descriptor loop.
- Identify the ES_IDs of the object descriptor, scene description and other streams described within the initial object descriptor.
- Acquire the set of all SL descriptors and FMC descriptors present in the second descriptor loop for any of the elementary_PIDs.
- Generate from these descriptors a stream map table between ES_IDs and associated elementary_PID plus FlexMux channel, if applicable.
- Locate the object descriptor stream using its ES_ID and the stream map table.
- Locate other streams described in the initial object descriptor using their ES_ID and the stream map table.
- Continuously monitor the object descriptor stream and identify ES_IDs of additional streams.
- Locate the additional streams using their ES_ID and the stream map table.

Figure R.2 gives an example of ISO/IEC 14496 program elements in a Transport Stream, consisting of object descriptor stream, scene description stream (BIFS-Command), BIFS-Anim and IPMP elementary streams. BIFS-Command and OD stream are conveyed by means of ISO_IEC_14496_sections, while BIFS-Anim and IPMP elementary streams are conveyed in PES packets referenced by two distinct elementary_PID values, without the use of the ISO/IEC 14496-1 FlexMux tool.

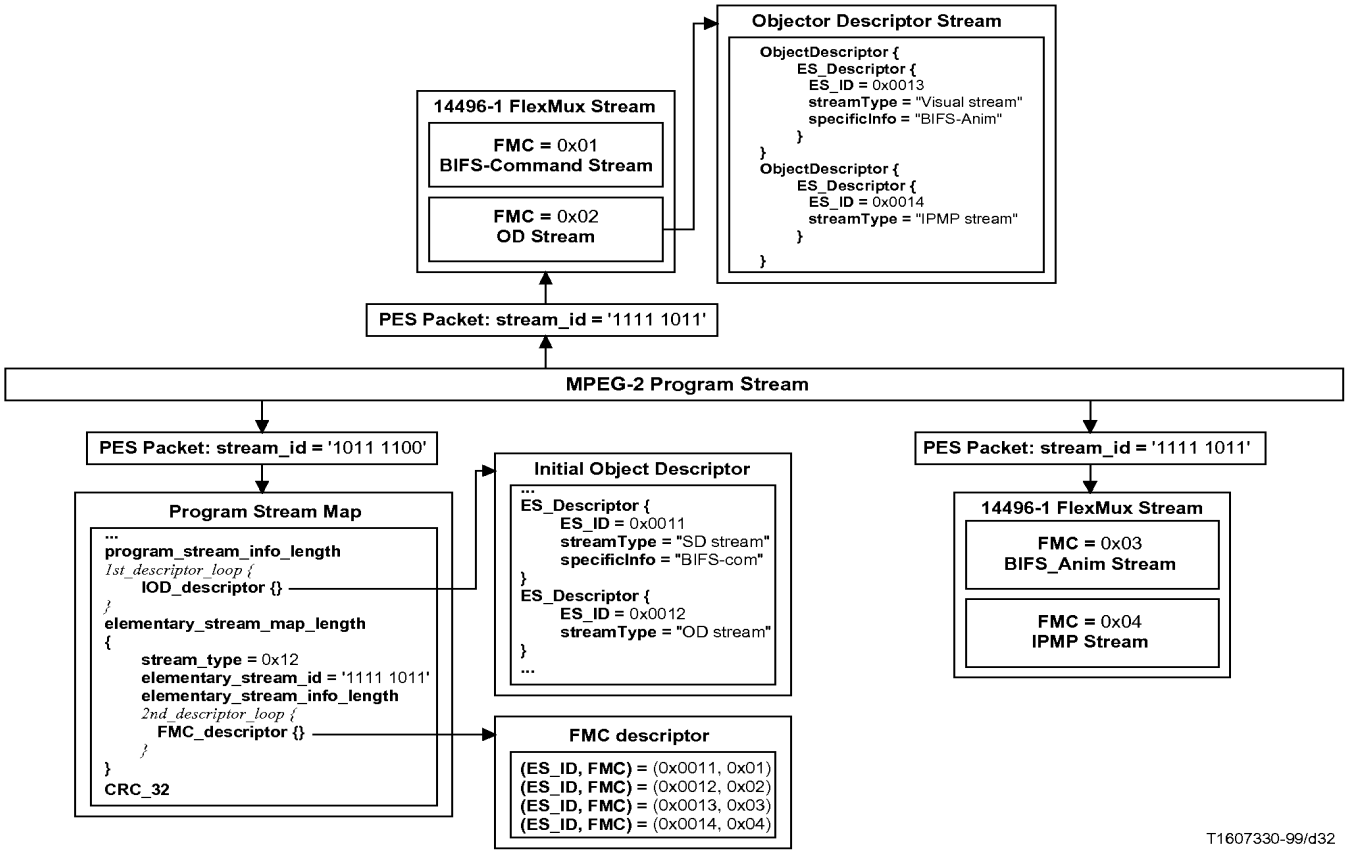


Figure R.1 – Example of ISO/IEC 14496 content in a Program Stream

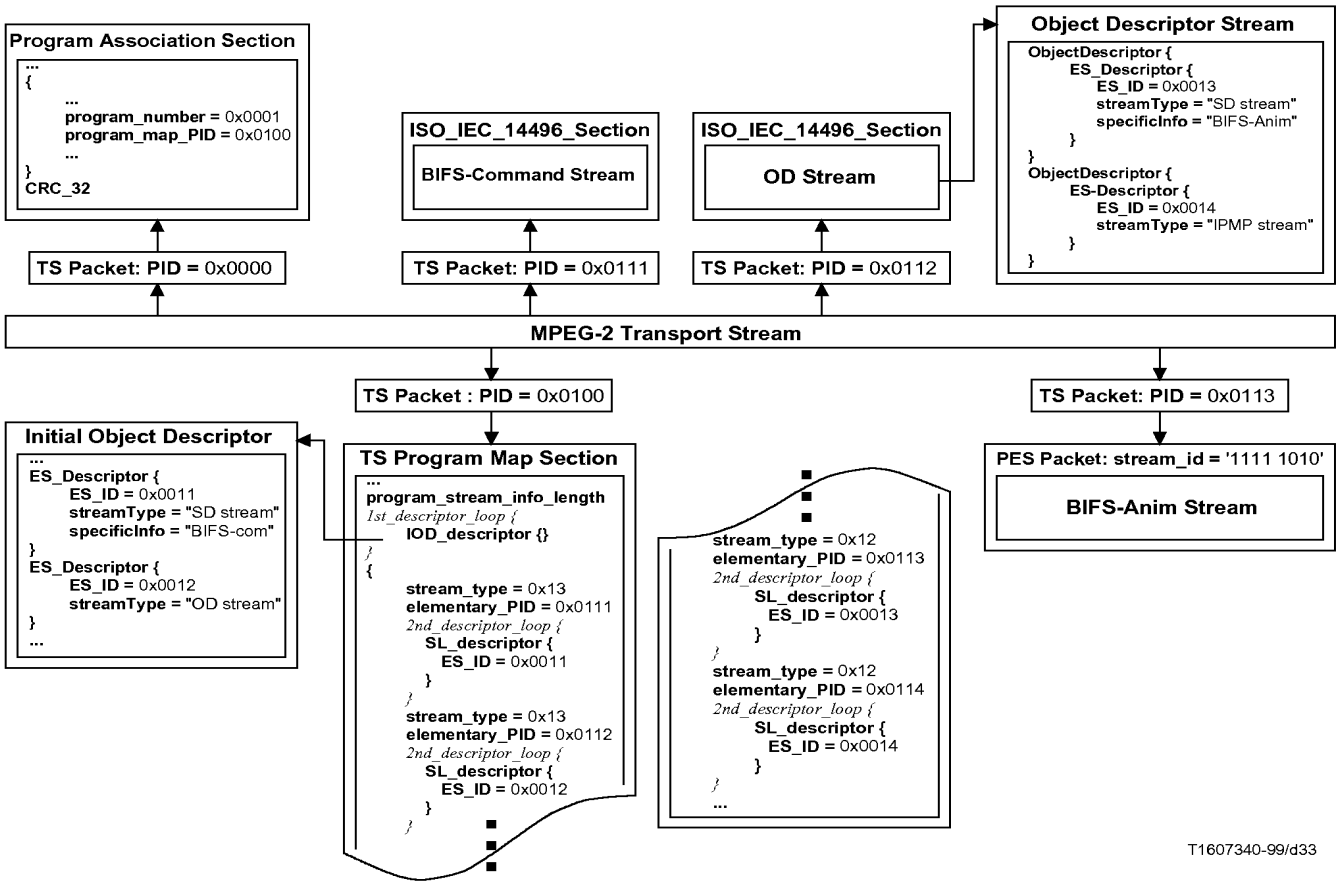


Figure R.2 – Example of ISO/IEC 14496 content in a Transport Stream

Annex S

Patents

(This annex does not form an integral part of this Recommendation | International Standard)

(This annex is only applicable to ISO/IEC 13818-1)

Patent List

The user's attention is called to the possibility that - for some of the processes specified in this Recommendation | International Standard conformance with this Recommendation | International Standard may require use of an invention covered by patent rights.

By publication of this part of Recommendation | International Standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. However, each company listed in this annex has undertaken to file with the Information Technology Task Force (ITTF) a statement of willingness to grant a license under such rights that they hold on reasonable and non-discriminatory terms and conditions to applicants desiring to obtain such a license.

Information regarding such patents can be obtained from the following organizations.

The table summarizes the formal patent statements received and indicates the parts of the standard to which the statement applies. Three "N"s in a row corresponding to a company means that the statement from the company did not mention any part. The list includes all organizations that have submitted informal patent statements. However, if no "X" is present, no formal patent statement has yet been received from that organization.

Table — List of companies supplying patent statements

Company	ISO/IEC 13818-1	ISO/IEC 13818-2	ISO/IEC 13818-3
AT&T	X	X	X
BBC Research Department		X	
Bellcore		X	
Belgian Science Policy Office	N	N	N
BOSCH	X	X	X
CCETT			
Columbia University in the City of New York	N	N	N
Compression Labs Incorporated	N	N	N
CSELT		X	
David Sarnoff Research Center	X	X	X
Deutsche Thomson-Brandt GmbH	N	N	N
France Telecom CNET	N	N	N
Fraunhofer Gesellschaft	N	N	N
FUJITSU Limited		X	
GC Technology Corporation	N	N	N
General Instruments	N	N	N
Goldstar	N	N	N
Hitachi, Ltd.	N	N	N
International Business Machines Corporation	X	X	X
IRT			X
KDD Co. Ltd.		X	
Massachusetts Institute of Technology	X	X	X
Matsushita Electric Industrial Co., Ltd.	N	N	N
Mitsubishi Electric Corporation	N	N	N
National Transcommunications Limited	N	N	N
NEC Corporation	N	N	N
Nippon Hoso Kyokai		X	
Nippon Telegraph and Telephone	N	N	N
Nokia Research Center		X	

Company	ISO/IEC 13818-1	ISO/IEC 13818-2	ISO/IEC 13818-3
Norwegian Telecom Research		X	
OKI Electric Industry Co. Ltd	N	N	N
Philips Consumer Electronics	N	N	N
Qualcomm Incorporated		X	
Royal PTT Nederland N.V., PTT Research (NL)			
Samsung Electronics	X	X	X
Scientific Atlanta	X	X	X
Siemens AG	N	N	N
Sharp Corporation	N	N	N
Sony Corporation	N	N	N
Texas Instruments	N	N	N
Thomson Consumer Electronics	N	N	N
Toshiba Corporation		X	
TV/Com	X	X	X
Victor Company of Japan Limited	X	X	X

ICS 35.040
Price based on 154 pages

© ISO/IEC 2000 – All rights reserved

**FEDERAL SERVICE FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADE MARKS
(ROSPATENT)**
30-1, Berezhkovskaya nab., 123995, Moscow Phone (499) 240-60-15, fax (495) 234-30-58

To No. **2412-136663RU/8132 of 06.10.2009**
(21) Our ref. **2006110321/09 (011234)**

To: NEVINPAT, P.O.Box 24,
191036, St. Petersburg
Attn. Polikarpov A.V.

May 05, 2010

DECISION TO GRANT A PATENT FOR AN INVENTION

(21) Application No.: **2006110321/09 (011234)** (22) Filing date: **30.03.2006**

As a result of substantive examination it has been established that

[ ] the claimed invention
[x] the claimed group of inventions

satisfies the requirements and patentability criteria set forth in the Civil Code of the Russian Federation. In view of the aforesaid the decision to grant a patent for invention is issued.

The decision issued on the basis of the substantive examination is enclosed.

Encl.: Decision on 7 pages in one copy

DECISION ISSUED ON THE BASIS OF EXAMINATION

(21) Application No.: **2006110321/09 (011234)** (22) Filing date: **30.03.2006**
(24) Date of beginning of the patent validity: **22.01.2003**
(85) Date of entering the national phase:

PRIORITY IS ESTABLISHED ON THE BASE OF

[ ] (22) filing date
[ ] (23) date of receipt of additional materials of
to an earlier application No. filed on
[x] (62) [x] priority date of application No. 2004125579 filed on 22.01.2003
from which said application is divided
[ ] filing date of application No filed on
from which said application is divided
[ ] (66) filing date of an earlier application No. filed on

[ ] (30) Primary application filing date

(31) Priority application No	(32) Priority date	(33) Country of priority
20020127	23.01.2002	FI

(86) PCT Application No. of
(87) International publication number and date of

(72) Inventor(s)
HANNUKSELA Miska, FI
(73) Patent owner(s)
Nokia Corporation, FI

(54) Title of the invention

GROUPING
OF IMAGE FRAMES IN VIDEO CODING

07 2	06.10.2009	092501

As a result of substantive examination conducted in relation to
[ ] the originally filed claims
[x] the claims amended by the Applicant

it has been established that

[ ] the invention
[x] the group of inventions

satisfies the requirements and patentability criteria set forth in Articles 1349 and 1350 of the Civil Code of the Russian Federation.

The accepted claims are presented on pages 3-7.

The original drawings and original disclosure will be published.

Encl.: Abstract corrected by the Examiner.

ФЕДЕРАЛЬНАЯ СЛУЖБА ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ
(РОСПАТЕНТ)
Бережковская наб., 30, корп. 1, Москва, Г-59, ГСП-5, 123995. Телефон (8-499) 240- 60- 15. Факс (8-495) 234- 30- 58

На № 2412-136663RU/8132 от 06.10.2009

Наш № 2006110321/09(011234)

При переписке просим ссылаться на номер заявки и сообщить дату получения настоящей корреспонденции от

0 5 МАЯ 2010

129090, Москва, ул. Б.Спасская, 25, стр.3,
ООО "Юридическая фирма Городисский и Партнеры",

пат.пов. Ю.Д.Кузнецову, рег.N 595

Р Е Ш Е Н И Е
о выдаче патента на изобретение

(21) Заявка № 2006110321/09(011234) (22) Дата подачи заявки 30.03.2006

В результате экспертизы заявки на изобретение по существу установлено, что

[ ] заявленное изобретение

[x] заявленная группа изобретений

относится к объектам патентных прав и соответствует условиям патентоспособности, предусмотренным Гражданским кодексом Российской Федерации, в связи с чем принято решение о выдаче патента на изобретение.

Заключение по результатам экспертизы прилагается.

Приложение: на 7л. в 1 экз.

Date G&P: 11/05/2010
0006095830

Руководитель Б.П.Симонов

ЗАКЛЮЧЕНИЕ ПО РЕЗУЛЬТАТАМ ЭКСПЕРТИЗЫ

(21) Заявка № 2006110321/09(011234) (22) Дата подачи заявки 30.03.2006

(24) Дата начала отсчета срока действия патента 22.01.2003

[ ] (85) Дата начала рассмотрения международной заявки на национальной фазе

ПРИОРИТЕТ УСТАНОВЛЕН ПО ДАТЕ

[ ] (22) подачи заявки

[ ] (23) поступления дополнительных материалов от
к ранее поданной заявке № от

[x] (62) [x] приоритета изобретения по первоначальной заявке № 2004125579 от 22.01.2003
из которой данная заявка выделена
[ ] подачи первоначальной заявки №
из которой данная заявка выделена

[ ] (66) подачи ранее поданной заявки № от

[x] (30) подачи первой(ых) заявки(ок) в государстве-участнике Парижской конвенции

(31) Номер первой(ых) заявки(ок)	(32) Дата подачи первой(ых) заявки(ок)	(33) Код страны
1. 20020127	23.01.2002	FI

[ ] (86) Заявка № PCT/ [ ] (96) Заявка № EA

[ ] (87) Номер публикации и дата публикации заявки PCT

(72) Автор(ы) ХАННУКСЕЛА Миска, FI

(73) Патентообладатель(и) НОКИА КОРПОРЕЙШН, FI

(54) Название изобретения ГРУППИРОВАНИЕ КАДРОВ ИЗОБРАЖЕНИЯ НА ВИДЕО КОДИРОВАНИИ

(см. на обороте)

07 2	ДОМ 06.10.2009	092501

ВНИМАНИЕ! *С целью исключения ошибок просьба проверить сведения, приведенные в заключении, т.к. они без изменения будут внесены в Государственный реестр изобретений Российской Федерации, и незамедлительно сообщить об обнаруженных ошибках.*

Адрес для переписки с патентообладателем или его представителем, который будет опубликован в официальном бюллетене указан на лицевой стороне решения

Адрес для направления патента указан на лицевой стороне решения

В результате экспертизы заявки по существу, проведенной в отношении уточненной заявителем формулы изобретения установлено соответствие заявленной группы изобретений требованиям статей 1349 и 1350 Гражданского кодекса Российской Федерации.

Формула изобретения приведена на странице(ах) 3-7.

Форма № 01а

(21) 2006110321/09

(51) МПК
H04N 7/50 (2006.01)

(57)

1. Способ кодирования видеопоследовательности, содержащей независимую последовательность кадров изображения, при этом по меньшей мере один опорный кадр изображения является прогнозируемым на основе по меньшей мере одного предыдущего кадра изображения, который предшествует в порядке декодирования предыдущему опорному кадру изображения, отличающийся тем, что содержит этап, на котором кодируют в видеопоследовательность указание по меньшей мере одного кадра изображения, который является первым, в порядке декодирования, кадром изображения упомянутой независимой последовательности, при этом упомянутый по меньшей мере один опорный кадр изображения включают в данную последовательность.

2. Способ по п. 1, отличающийся тем, что кодируют упомянутое указание в видеопоследовательность в качестве отдельного флага, включаемого в заголовок среза.

3. Способ по п.1, отличающийся тем, что кодируют значения идентификаторов для кадров изображения согласно схеме нумерации,

переустанавливают значение идентификатора для указанного первого кадра изображения независимой последовательности.

4. Способ по п.1, отличающийся тем, что кодируют в видеопоследовательность значение идентификатора для независимой последовательности.

5. Видеокодер для кодирования видеопоследовательности, содержащей независимую последовательность кадров изображения, при этом по меньшей мере один опорный кадр изображения является прогнозируемым на основе по меньшей мере одного предыдущего кадра изображения, который предшествует в

порядке декодирования предыдущему опорному кадру изображения, отличающийся тем, что выполнен с возможностью кодирования в видеопоследовательность указания по меньшей мере одного кадра изображения, который является первым, в порядке декодирования, кадром изображения упомянутой независимой последовательности, при этом упомянутый по меньшей мере один опорный кадр изображения включается в данную последовательность.

6. Видеокодер по п. 5, отличающийся тем, что выполнен с возможностью кодирования упомянутого указания в видеопоследовательность в качестве отдельного флага, включаемого в заголовок среза.

7. Видеокодер по п. 5, отличающийся тем, что выполнен с возможностью кодирования значений идентификаторов для кадров изображения согласно схеме нумерации, переустановки значения идентификатора для указанного первого кадра изображения независимой последовательности.

8. Видеокодер по п. 5, отличающийся тем, что выполнен с возможностью кодирования в видеопоследовательность значения идентификатора для независимой последовательности.

9. Машиночитаемый носитель, на котором хранится исполняемый в устройстве обработки данных компьютерный программный код, предназначенный для кодирования видеопоследовательности, содержащей независимую последовательность кадров изображения, при этом по меньшей мере один опорный кадр изображения является прогнозируемым на основе по меньшей мере одного предыдущего кадра изображения, который предшествует в порядке декодирования предыдущему опорному кадру изображения, отличающийся тем, что компьютерный программный код содержит компьютерный программный код для кодирования в видеопоследовательность указания по меньшей мере одного кадра изображения, который является первым, в порядке декодирования, кадром изображения упомянутой независимой последовательности, при этом упомянутый по меньшей мере один опорный кадр изображения включается в данную последовательность.

10. Способ декодирования сжатой видеопоследовательности, при этом по меньшей мере один опорный кадр изображения является прогнозируемым на основе по меньшей мере одного предыдущего кадра изображения, который предшествует в порядке декодирования предыдущему опорному кадру изображения, отличающийся тем, что содержит этапы, на которых декодируют из видеопоследовательности указание по меньшей мере одного кадра изображения, который является первым, в порядке декодирования, кадром изображения независимой последовательности, причем данная последовательность включает в себя упомянутый по меньшей мере один опорный кадр изображения, начинают декодирование видеопоследовательности с упомянутого первого кадра изображения упомянутой независимой последовательности, при этом видеопоследовательность декодируется без прогнозирования на основе какого-либо кадра изображения, декодированного до упомянутого первого кадра изображения.

11. Способ по п. 10, отличающийся тем, что упомянутое указание является отдельным флагом, включенным в заголовок среза.

12. Способ по п. 10 или 11, отличающийся тем, что декодируют значения идентификаторов для кадров изображения согласно схеме нумерации, переустанавливают значение идентификатора для указанного первого кадра изображения независимой последовательности.

13. Видеодекодер для декодирования сжатой видеопоследовательности, при этом по меньшей мере один опорный кадр изображения является прогнозируемым на основе по меньшей мере одного предыдущего кадра изображения, который предшествует в порядке декодирования предыдущему опорному кадру изображения, отличающийся тем, что выполнен с возможностью декодировать из видеопоследовательности указание по меньшей мере одного кадра изображения, который является первым, в порядке декодирования, кадром изображения независимой последовательности, причем данная последовательность включает в себя упомянутый по меньшей мере один опорный кадр изображения, начинать декодирование видеопоследовательности с

упомянутого первого кадра изображения упомянутой независимой последовательности, при этом видеопоследовательность декодируется без прогнозирования на основе какого-либо кадра изображения, декодированного до упомянутого первого кадра изображения.

14. Машиночитаемый носитель, на котором хранится исполняемый в устройстве обработки данных компьютерный программный код, предназначенный для декодирования сжатой видеопоследовательности, при этом по меньшей мере один опорный кадр изображения является прогнозируемым на основе по меньшей мере одного предыдущего кадра изображения, который предшествует в порядке декодирования предыдущему опорному кадру изображения, отличающийся тем, что компьютерный программный код содержит компьютерный программный код для того, чтобы декодировать из видеопоследовательности указание по меньшей мере одного кадра изображения, который является первым, в порядке декодирования, кадром изображения независимой последовательности, причем данная последовательность включает в себя упомянутый по меньшей мере один опорный кадр изображения, компьютерный программный код для того, чтобы начинать декодирование видеопоследовательности с упомянутого первого кадра изображения упомянутой независимой последовательности, при этом видеопоследовательность декодируется без прогнозирования на основе какого-либо кадра изображения, декодированного до упомянутого первого кадра изображения.

15. Видеосигнал, содержащий сжатую видеопоследовательность, включающую в себя независимую последовательность кадров изображения, при этом по меньшей мере один опорный кадр изображения является прогнозируемым на основе по меньшей мере одного предыдущего кадра изображения, который предшествует в порядке декодирования предыдущему опорному кадру изображения, отличающийся тем, что содержит указание по меньшей мере одного кадра изображения, который является первым, в порядке декодирования, кадром изображения упомянутой независимой

последовательности, при этом упомянутый по меньшей мере один опорный кадр изображения включен в данную последовательность.

16. Видеосигнал по п. 15, отличающийся тем, что содержит значение идентификатора для упомянутой независимой последовательности.

(56)

STEPHAN WENGER, Temporal Scalability using P-Pictures for Low-Latency Applications, IEEE Signal Processing Society 1998 Workshop on Multimedia Signal Processing, Los Angeles, December 7-9, 1998

RU 2128405 C1, 1999.03.27

WO0147283 A1, 2001.06.28

US 6072831 A, 2000.06.06

US 6337881 B1, 2002.01.08

RU 2137194 C1, 1999.09.10

Information technology - Generic coding of moving pictures and associated audio information: Systems, ISO/IEC 13818-1, Second edition, 2000.12.01

ILLGNER K. et al, SPATIALLY SCALABLE VIDEO COMPRESSION EMPLOYING RESOLUTION PYRAMIDS, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, New York, vol.15, no.9, 01.12.1997, c.c.1688-1703

При публикации сведений о выдаче патента будут использованы первоначальные чертежи и описание.

Приложение: 1.Реферат, скорректированный экспертизой, на 1 л. в 1 экз.

Заместитель заведующего
отделом электрорадиотехники ФГУ ФИПС С.В. Махотина

В.Н.Воропай 8 (495) 730 76 49

К заявке № 2006110321/09

(54) ГРУППИРОВАНИЕ КАДРОВ ИЗОБРАЖЕНИЯ НА ВИДЕО КОДИРОВАНИИ

Реферат

(57) Изобретение относится к области видеокодирования, и в частности, к группированию видео файлов с использованием потоковой передачи. Техническим результатом является обеспечение высокого качества восстановленного изображения. Техническим результат достигается тем, что способ кодирования видеопоследовательности, содержащей независимую последовательность кадров изображения, при этом по меньшей мере один опорный кадр изображения является прогнозируемым на основе по меньшей мере одного предыдущего кадра изображения, который предшествует в порядке декодирования предыдущему опорному кадру изображения. В видеопоследовательность кодируется указание по меньшей мере одного кадра изображения, который является первым, в порядке декодирования, кадром изображения упомянутой независимой последовательности, при этом упомянутый по меньшей мере один опорный кадр изображения включается в данную последовательность. На стадии декодирования из видеопоследовательности декодируется указание по меньшей мере одного кадра изображения, и декодирование видеопоследовательности начинается с упомянутого первого кадра изображения упомянутой независимой последовательности, при этом видеопоследовательность декодируется без прогнозирования на основе какого-либо кадра изображения, декодированного до упомянутого первого кадра изображения. 7 н.п., 9 з.п. ф-лы, 14 ил.

Референт В.Н. Воропай

(19) **RU** (11) **2 137 194** (13) **C1**

(51) МПК[6] **G 06 K 9/00, G 06 T 7/20**

РОССИЙСКОЕ АГЕНТСТВО
ПО ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ

(21), (22) Заявка: 98114418/09, 15.07.1998

(24) Дата начала действия патента: 15.07.1998

(46) Дата публикации: 10.09.1999

(56) Ссылки: K.R.RAO, J.J.HWANG, "Techniques and Standards for Image, Video and Audio Coding", 1996, PRENTICE-HALL PTR, ISBN 0-13-309907-5. SU 453978 A, 07.03.84. SU 802974 A, 07.02.81. EP 0647919 A1, 12.04.95. GB 2282021 A, 22.03.95.

(98) Адрес для переписки:
103064, Москва, ул.Казакова, 16,
НИИР-Канцелярия "Патентные поверенные Квашнин, Сапельников и Партнеры" Патентному поверенному Сапельникову Д.А.

(71) Заявитель:
Дворкович Александр Викторович,
Дворкович Виктор Павлович,
Зубарев Юрий Борисович,
Соколов Александр Юрьевич

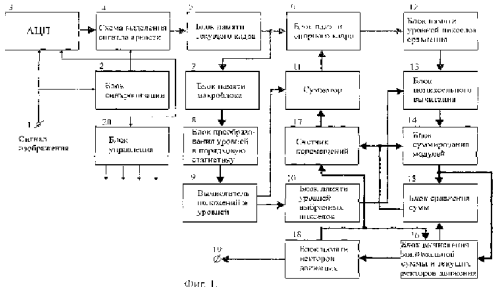
(72) Изобретатель: Дворкович А.В.,
Дворкович В.П., Зубарев Ю.Б., Соколов А.Ю.

(73) Патентообладатель:
Дворкович Александр Викторович,
Дворкович Виктор Павлович,
Зубарев Юрий Борисович,
Соколов Александр Юрьевич

(54) СПОСОБ АНАЛИЗА ВЕКТОРОВ ДВИЖЕНИЯ ДЕТАЛЕЙ В ДИНАМИЧЕСКИХ ИЗОБРАЖЕНИЯХ

(57) Реферат:

Изобретение относится к видеоинформационной технике и может найти применение при разработке цифровых кодирующих устройств для видеотелефонии, видеоконференцсвязи, телевизионного цифрового вещания стандартной и высокой четкости. Технический результат заключается в сокращении количества вычислительных операций, увеличении зоны поиска векторов движения деталей в динамических изображениях и, вследствие этого, повышении качества воспроизведения быстро перемещающихся деталей. Технический результат достигается за счет того, что в способе анализа векторов движения деталей в динамических изображениях все множество пикселов каждого из макроблоков изображения разбивается на участки, в каждом из которых выбирается только один пиксел таким образом, что уровни всей совокупности выбранных пикселов характеризуют изменение рельефа сигнала яркости макроблока текущего кадра, и затем контрольная сумма, являющаяся суммой норм попиксельной разности уровней в текущем и опорном кадрах, вычисляется лишь по выбранным упомянутым пикселам. 1 з.п. ф-лы, 8 ил., 4 табл.



Фиг. 1.

RU 2137194 C1

RU 2137194 C1

(19) **RU** (11) **2 137 194** (13) **C1**

(51) Int. Cl.[6] **G 06 K 9/00, G 06 T 7/20**

RUSSIAN AGENCY
FOR PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION

(21), (22) Application: **98114418/09, 15.07.1998**

(24) Effective date for property rights: **15.07.1998**

(46) Date of publication: **10.09.1999**

(98) Mail address:
**103064, Moskva, ul.Kazakova, 16,
NIIR-Kantseljarija "Patentnye poverennye
Kvashnin, Sapel'nikov i Partnery" Patentnomu
poverennomu Sapel'nikovu D.A.**

(71) Applicant:
**Dvorkovich Aleksandr Viktorovich,
Dvorkovich Viktor Pavlovich,
Zubarev Jurij Borisovich,
Sokolov Aleksandr Jur'evich**

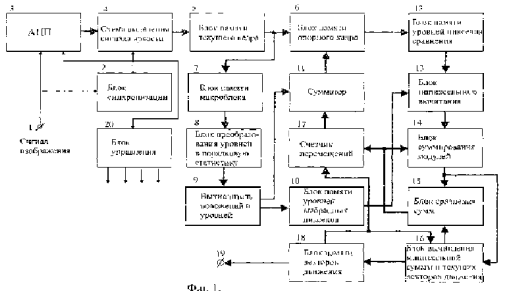
(72) Inventor: **Dvorkovich A.V.,
Dvorkovich V.P., Zubarev Ju.B., Sokolov A.Ju.**

(73) Proprietor:
**Dvorkovich Aleksandr Viktorovich,
Dvorkovich Viktor Pavlovich,
Zubarev Jurij Borisovich,
Sokolov Aleksandr Jur'evich**

(54) METHOD FOR ANALYSIS OF MOVEMENT VECTORS OF ELEMENTS IN DYNAMIC IMAGES

(57) Abstract:

FIELD: video presentation devices, in particular, digital encoders for video telephones, conference calls, standard and high-definition TV systems. SUBSTANCE: method involves splitting complete set of pixels of each large image block into regions in which only one pixel is chosen so that levels of complete set of chosen pixels define alteration of shape of brightness signal of large image block of current frame. Then check sum which is defined as sum of norms of by-pixel difference in levels in current and reference frames is calculated using data from only chosen pixels. EFFECT: decreased amount of calculations, increased region for search of movement vectors of elements, improved quality of representation of fast moving elements. 2 cl, 13 dwg, 4 tbl



RU 2137194 C1

RU 2137194 C1

Изобретение относится к видеоинформационной технике и может найти применение при разработке цифровых кодирующих устройств для видеотелефонии, видеоконференцсвязи, телевизионного цифрового вещания стандартной и высокой четкости.

Известно множество способов анализа векторов движения деталей в динамических изображениях.

Наиболее простым и точным является способ поиска векторов движения макроблоков на основе полного перебора [1, с. 89-91]. Для поиска вектора движения $\vec{V} = (V_x, V_y)$ рассматривается норма разницы сигналов яркости двух макроблоков в текущем и опорном кадрах SAD со сдвигом на вектор движения:

$$SAD = \sum_{x,y=1,16} |F(x,y,t) - F(x-V_x, y-V_y, t-1)| \quad (1)$$

Под опорным подразумевается кадр, относительно которого осуществляется поиск векторов движения макроблоков текущего кадра. Иногда в качестве суммы норм принимают величину

$$SAD = \sum_{x,y=1,16} [F(x,y,t) - F(x-V_x, y-V_y, t-1)]^2 \quad (1.1)$$

Здесь F - значение яркости, (x, y) - пространственные координаты точки в кадре, t - временной индекс кадра, суммирование производится по всем точкам макроблока. Значение $\vec{V}$, для которого норма SAD имеет наименьшее значение, принимается за искомый вектор. Вектора движения ищутся методом полного перебора в некоторой ограниченной окрестности: $\min < V_x, V_y < \max$. Предполагая, что размер этой окрестности равен $\pm N$ по координатам x и y, получим для числа операций, необходимых для определения вектора движения одного макроблока, величину порядка $3 \cdot 256 (2N+1)^2$. На один пиксел макроблока количество операций составляет $3 \cdot (2N+1)^2$, что уже при N = 15 (значения векторов движения в пределах ± 15 точек) составляет значительную величину более 10^3 операций / пиксел.

Этот способ обычно используется в качестве эталонного для оценки качества других способов анализа векторов движения деталей в динамических изображениях.

Недостатком этого способа является требуемое большое количество вычислительных операций и в связи с этим низкое быстродействие.

Для уменьшения количества операций были предложены различные способы и методы, основанные на ограничении возможного числа кандидатов на вектора движения, в частности логарифмический и иерархический методы поиска векторов движения [1, с.92-96]. При логарифмическом поиске вектора движения поиск осуществляется в небольшой окрестности с шагом изменения значения вектора движения в 2 пиксела до тех пор, пока не будет найден локальный минимум SAD, после чего значение вектора движения уточняется с шагом в один пиксел в окрестности этого минимума. В 3-х шаговом способе поиск ведется первоначально по всей окрестности с шагом в 4 пиксела, затем вокруг найденного минимума с шагом в 2 пиксела, и, наконец, с шагом в один пиксел. В иерархическом способе первоначально проводится поиск векторов движения для специальным образом передискретизированного изображения, обладающего более грубым разрешением, потом значения найденных векторов движения уточняются при переходе к следующему уровню более высокого разрешения.

Недостатками этих ускоренных способов анализа являются существенные ошибки анализа движений деталей в динамических изображениях вследствие достаточно сложной зависимости SAD от V, которая практически всегда не является гладкой функцией. Функция $SAD(\vec{V})$ может иметь несколько экстремумов, что усложняет применение и значительно снижает эффективность данных методов.

Наиболее близким по технической сущности к заявляемому техническому решению является наиболее простой и точный способ [1] анализа векторов движения деталей в динамических изображениях, включающий преобразование последовательности кадров изображений в цифровую форму, запоминание дискретных отсчетов яркости текущего и соседнего по времени кадров, разбиение текущего кадра на макроблоки и поиск вектора движения каждого из макроблоков текущего кадра относительно опорного кадра посредством минимизации по рассматриваемому множеству векторов движения контрольной суммы данного макроблока, являющейся суммой норм попиксельной разности уровней в текущем и опорном кадрах.

Как уже было сказано, недостатком этого способа является требуемое большое количество вычислительных операций и в связи с этим низкое быстродействие. Этот недостаток не позволяет реально обеспечить большие зоны анализа движения, что приводит к снижению качества воспроизведения быстродвижущихся деталей в динамических изображениях.

В основу изобретения положена задача за счет сокращения объема вычислительных операций повысить быстродействие кодирующих устройств и производить анализ движения деталей динамических изображений в больших областях, сократив при этом объем сжатой информации и повысив качество воспроизведения быстродвижущихся деталей.

Эта задача решена посредством способа анализа векторов движения деталей в динамических изображениях, включающего преобразование последовательности кадров изображений в цифровую форму, запоминание дискретных отсчетов яркости текущего и опорного кадров, разбиение текущего кадра на макроблоки и поиск вектора движения каждого из макроблоков текущего кадра относительно опорного кадра посредством минимизации по рассматриваемому множеству векторов движения контрольной суммы данного макроблока, являющейся суммой норм попиксельной разности уровней в текущем и опорных кадрах. Согласно изобретению, все

RU 2137194 C1

RU 2137194 C1

множество пикселов рассматриваемого макроблока разбивают на участки, в каждом из которых определенным образом выбирают только один пиксел, и упомянутую контрольную сумму вычисляют лишь по выбранным упомянутым пикселам, при этом упомянутые пикселы в каждом из участков выбирают таким образом, что их уровни в смежных участках наиболее отличаются друг от друга.

Было бы целесообразно при использовании описанного выше способа анализа векторов движения деталей в динамических изображениях дополнительно для каждого текущего рассматриваемого значения вектора движения вычисление контрольной суммы по выбранным упомянутым пикселам производить в порядке убывания отклонения значения сигнала в каждом пикселе от среднего значения по всей совокупности упомянутых пикселов и прерыванием дальнейшего вычисления контрольной суммы в случае, когда она превышает значение минимальной контрольной суммы, найденной среди всех уже рассмотренных векторов движения.

На фиг. 1 изображен один из возможных вариантов устройства для реализации способа анализа векторов движения деталей в динамических изображениях; на фиг. 2 - 8 - диаграммы, поясняющие его работу.

Устройство, изображенное на фиг. 1, содержит подключенные к входу 1 параллельно блок синхронизации 2 и последовательно соединенные аналого-цифровой преобразователь 3, схему выделения 4 сигнала яркости, последовательно соединенные блок памяти 5 текущего кадра и блок памяти 6 опорного кадра, подключенные к выходам блока памяти текущего кадра последовательно соединенные блок памяти 7 текущего макроблока, блок преобразования 8 в порядковую статистику уровней пикселов макроблока и вычислитель 9 положений и уровней характерных пикселов макроблока, первые выходы которого соединены с блоком памяти 10 уровней выбранных пикселов, а вторые входы - через сумматор 11 к входам управления блока памяти опорного кадра 6, выходы которого через блок памяти 12 уровней пикселов сравнения подключены к первым входам блока 13 попиксельного вычитания уровней характерных пикселов текущего и опорного кадров, вторые входы которого подключены к вторым выходам блока памяти 10, а выходы - к блоку суммирования 14 модулей, выходы которого параллельно подключены к блоку сравнения сумм 15 и блоку вычисления 16 минимальной суммы и текущих векторов движения, вторые входы блока сравнения сумм 15 подключены к выходам блока 16, а выходы - непосредственно ко входам управления блока 14 и через счетчик перемещений 17 к второму входу сумматора 11, вторые выходы блока 16 подключены через блок памяти 18 векторов движения к выходам устройства 19, вторые выходы блока 18 соединены параллельно с дополнительными входами блока 16 и счетчика 17, выходы блока синхронизации 2 соединены с входами управления блоков 3 и 4, а также через блок управления 20 - с входами синхронизации работы блоков 5-18 устройства.

Сущность предполагаемого способа состоит в следующем.

Цель проводимых над сигналами динамического изображения преобразований состоит в резком сокращении объема вычислительных операций при увеличении зоны поиска векторов движения деталей в динамических изображениях, что позволяет существенно повысить быстродействие кодирующих устройств и производить анализ движения деталей динамических изображений в больших областях, сократив при этом объем сжатой информации и повысив качество воспроизведения быстродвижущихся деталей.

Изложение предлагаемого способа иллюстрируется на примере анализа вектора движения одного из макроблоков динамического изображения, приведенного на фиг. 2: фиг. 2а - текущий кадр, фиг. 2б - опорный кадр. На фиг. 3а приведена в увеличенном масштабе часть изображения текущего кадра в увеличенном масштабе с обведенным макроблоком, вектор движения которого определяется в зоне опорного кадра, изображенной на фиг. 3б. Макроблок имеет размер 16•16 пикселов, зона поиска движения - 64•64 пиксела. На фиг. 3б обозначено положение смещенного макроблока в опорном кадре.

В случае использования стандартного метода поиска вектора движения в указанной зоне поиска требуется использование вычислительной мощности в $3 \cdot 49^2 = 7203$ операций на каждый пиксел макроблока. Рельеф сигнала яркости макроблока изображен на фиг. 4а, а на фиг. 4б приведены цифровые значения пикселов этого макроблока.

В соответствии с предлагаемым способом макроблок разбивается на несколько участков, например на девять, как показано на фиг. 4б. В каждом из этих участков выбирается лишь один пиксел.

Один из вариантов выбора может быть связан с использованием порядковых статистик. В этом случае все значения пикселов $F_1 \ldots F_{256}$ располагаются в порядке возрастания уровней, как показано в таблице на фиг. 5. В этой таблице наряду со значениями уровней пикселов $F_{(1)} \le F_{(2)} \le \ldots F_{(255)} \le F_{(246)}$ приведена принадлежность их к соответствующим выбранным девяти участкам.

Для того, чтобы на точность оценки векторов движения в меньшей степени влияло наличие помех в сигнале яркости, в качестве первых двух пикселов принимаются пикселы, уровни в которых соответствуют следующему за минимальным и предшествующий максимальному (в данном случае - пиксел с уровнем $F_{(2)}$= 24, расположенный на 5-м участке макроблока и пиксел с уровнем $F_{(255)}$=236, расположенный на 7-м участке макроблока). Если эти пикселы оказываются расположенными в одном участке, то положение одного, например $F_{(2)}$, - фиксируется, и пикселы $F_{(254)}$, $F_{(253)}$ ... перебираются, пока участок окажется не тем, в котором располагается пиксел с уровнем $F_{(2)}$.

Пикселы в других участках выбираются отступая от $F_{(2)}$ в сторону увеличения индекса

RU 2137194 C1

RU 2137194 C1

и от $F_{(255)}$ в сторону уменьшения индекса, например, на равное количество пикселов $\frac{256}{9} \approx 28$. Следовательно, выбранными пикселами должны быть $F_{(30)}$=87, расположенный на 4-м участке и $F_{(227)}$=178, расположенный на 6-м участке.

Аналогично следующими пикселами выбираются $F_{(59)}$=112, расположенный на 9-ом участке, и $F_{(198)}$=164, расположенный на 1-м участке.

Далее выбираются пиксел с уровнем $F_{(88)}$=124, расположенный на 8-м участке, и пиксел с уровнем $F_{(169)}$=151, расположенный на 5-м участке. Учитывая, что на 5-м участке пиксел уже был выбран ранее ($F_{(2)}$), выбор осуществляется из двух пикселов $F_{(170)}$ и $F_{(168)}$ в данном случае с равными уровнями. Из них выбирается пиксел $F_{(170)}$, расположенный на 3-м участке.

И, наконец, в качестве претендента на выбранное значение пиксела можно указать $F_{(117)}$= 132. Но этот пиксел оказался также расположенным на 6-м участке и по этой причине подбирается пиксел с тем же уровнем $F_{(118)}$, расположенный на 2-м участке.

Таким образом,

на 1-м участке выбран пиксел с уровнем 164;

на 2-м участке выбран пиксел с уровнем 132;

на 3-м участке выбран пиксел с уровнем 151;

на 4-м участке выбран пиксел с уровнем 87;

на 5-м участке выбран пиксел с уровнем 24;

на 6-м участке выбран пиксел с уровнем 178;

на 7-м участке выбран пиксел с уровнем 236;

на 8-м участке выбран пиксел с уровнем 124;

на 9-м участке выбран пиксел с уровнем 112.

Для реализации этого метода подбора пикселов требуется использование примерно $\log_2 256 = 8$ вычислительных операций на каждый пиксел макроблока.

Возможен более простой метод выбора характерных пикселов в каждом из участков макроблока, например, путем чередования максимальных и минимальных значений уровней в смежных участках. Предположим, на заштрихованных на фиг. 4б участках выбираются пикселы с минимальными значениями уровней, а на остальных - с максимальными значениями уровней. Тогда

на 1-м участке будет выбран пиксел с уровнем 81;

на 2-м участке будет выбран пиксел с уровнем 196;

на 3-м участке будет выбран пиксел с уровнем 84;

на 4-м участке будет выбран пиксел с уровнем 221;

на 5-м участке будет выбран пиксел с уровнем 24;

на 6-м участке будет выбран пиксел с уровнем 237;

на 7-м участке будет выбран пиксел с уровнем 61;

на 8-м участке будет выбран пиксел с уровнем 188;

на 9-м участке будет выбран пиксел с уровнем 94.

Этот более простой метод подбора пикселов требует использования лишь одной вычислительной операции на пиксел.

Для поиска вектора движения $\vec{V} = (V_x, V_y)$ в предлагаемом способе рассматривается норма разницы сигналов яркости выбранных точек двух макроблоков в текущем и опорном кадрах SAD1 со сдвигом на вектор движения:

$$SAD1 = \sum_{\substack{(x,y)-\text{координаты выбранных точек в соответствующих участках макроблока}} |F(x,y,t)-F(x-V_x,y-V_y,t-1)| \quad (2)$$

Очевидно, что скорость анализа движения при использовании предлагаемого способа увеличивается в N/M раз, где N - число пикселов в макроблоке, M - количество участков, на которое поделен макроблок. В данном случае скорость анализа увеличивается в 256/9=28,44 раз.

Дополнительно скорость анализа движения макроблоков увеличивается в соответствии с п. 2 настоящего изобретения за счет определенного подбора вычисления контрольной суммы SAD 1.

В этом случае вычисляется среднее значение яркости выбранных пикселов макроблока

$$F_{ср} = \frac{1}{M} \sum_{\substack{(x,y)-\text{координаты выбранных точек в соответствующих участках макроблока}} F(x,y,t) \quad (3)$$

Затем вычисляются модули разностей $X(x,y) = |F(x,y,t)-F_{ср}|$,

где (x, y) - координаты выбранных точек в соответствующих участках макроблока, и контрольная сумма (2) вычисляется последовательно по координатам (x, y), для которых величины X(x, y) располагаются в убывающей последовательности.

В случае, если текущий выбранный вектор $\vec{V} = (V_x, V_y)$ не соответствует истинному вектору движения, для которого контрольная сумма минимальна, рассчитываемая контрольная сумма статистически достаточно быстро возрастает и в этом случае обычно не требуется производить вычисления по всем выбранным точкам макроблока. При этом в среднем по кадру скорость анализа движения макроблоков дополнительно возрастает в два раза.

Эффективность предлагаемого способа анализа движения иллюстрируется на фиг. 6-8.

На фиг. 6 приведены изображения разности текущего и опорного кадров после компенсации движения (а) и векторов движения (б), рассчитанных стандартным методом с использованием всех 256 пикселов макроблока.

На фиг. 7 приведены изображения разности текущего и опорного кадров после компенсации движения (а) и векторов

RU 2137194 C1

RU 2137194 C1

движения (б), рассчитанных предлагаемым способом с использованием лишь 9 пикселов макроблока.

На фиг. 8 приведены значения результирующей контрольной суммы до и после проведения анализа векторов движения; цифрой 1 обозначена кривая, характеризующая контрольную сумму до проведения анализа движения (без компенсации движения), цифрой 2 - кривая, характеризующая контрольную сумму при стандартном медленном анализе движения макроблоков с использованием 256 пикселов; цифрой 3 - кривая, характеризующая контрольную сумму при предложенном быстром способе анализа движения макроблоков с использованием лишь 9 пикселов.

Из визуального сравнения диаграмм фиг.6б и 7б видно, что анализ движения макроблоков с использованием стандартного и предлагаемого способов практически во всех областях кадра, насыщенных резко изменяющимся рельефом, осуществляется одинаково. Отличия возникают лишь в областях с малым изменением рельефа. Это подтверждается фиг. 8, на которой видно, что различие контрольных сумм оказывается малым (менее 10%).

Испытания различных динамических изображений показывают, что при увеличении числа выбранных участков, скажем, с 9 до 16 различие контрольных сумм с использованием стандартного и предлагаемого способа практически отсутствует.

Устройство для реализации предложенного способа работает следующим образом.

Предположим, на вход 1 устройства поступает аналоговый сигнал изображения, например полный цветовой телевизионный сигнал стандартных систем SECAM, PAL или NTSC. Со входа этот сигнал подается параллельно на блок синхронизации 2, выделяющий соответствующие сигналы и обеспечивающий формирование импульсов дискретизации, и на аналого-цифровой преобразователь АЦП 2, в котором дискретные отсчеты сигнала преобразуются в цифровой код, поступающий на схему выделения сигнала яркости 4, устраняющую цветовые поднесущие из полного цветового телевизионного сигнала. Выделение сигнала яркости необходимо, поскольку в соответствии со стандартами MPEG анализ движения деталей изображения осуществляется только при использовании яркостной компоненты изображения. Синхронизация работы схемы 4 осуществляется также дискретизирующими импульсами, поступающими с блока синхронизации 2.

Цифровой поток сигнала яркости со схемы 4 последовательно поступает на блок памяти 5 текущего кадра и блок памяти 6 опорного (предыдущего) кадра, в которых запоминаются дискретные отсчеты сигналов яркости текущего кадра, анализ движения макроблоков которого осуществляется относительно соответствующих структур опорного кадра.

Выходы блока 5 соединены со входами блока памяти макроблока 7, анализ движения которого производится. В этом блоке запоминаются значения рельефа сигнала яркости макроблока 16 •16 пикселов. После вычисления вектора движения 1-го макроблока, соответствующего левому верхнему углу изображения, в память этого блока вводится рельеф сигнала яркости следующего за ним макроблока. Отсчет номера макроблока осуществляется обычно слева направо и сверху вниз.

После запоминания уровней яркости всех пикселов макроблока они в блоке 8 преобразуются в порядковую статистику.

Обычно такое преобразование производится с использованием быстрого алгоритма, требующего для N чисел $N\log_2 N$ операций. При построении порядковой статистики одновременно фиксируются координаты пикселов, которым соответствует каждое их значение.

По значениям порядковой статистики в соответствии с упомянутым выше способом с помощью вычислителя положений и уровней 9 выбираются расположенные в выбранных участках по одному пикселу, по значениям, например, номеров порядковых статистик, приблизительно равноудаленных друг от друга. Принцип расчета поясняется для выбранного макроблока фиг.4а с помощью таблиц на фиг. 4б и 5. Рассчитанные выбранные пикселы соответствуют координатам, приведенным в табл. 1.

В случае использования описанного выше второго варианта выбора точек их координаты будут иными (см. табл. 2)

Таким образом, в вычислителе 9 выделяются значения уровней яркости выбранных пикселов и их координаты x и y.

В этом же блоке осуществляется преобразование выбранных пикселов в последовательность по мере убывания отклонений значений уровней пикселов от их среднего значения, для этой цели в первом варианте вычисляется среднее значение уровней

$F_{ср}$ =(164+132+151+87+24+178+236+124+112)/9 ≈ 134, а затем модули

$X(i) = X(x,y) = |F(x,y,t)-F_{ср}|$:

X(1) = 30; X(2) = 2; X(3) = 17; X(4)= 47; X(5)= 110;

X(6) = 44; X(7) = 102; X(8) = 10; X(9) = 22.

При этом выбранные пикселы представляются и запоминаются в вычислителе 9 в последовательности уменьшения модулей разностей (см. табл. 3).

При выборе пикселов по второму варианту среднее значение яркости будет равно:

$F_{ср}$ =(81+196+84+221+24+237+61+188+94)/9≈132.

Модули разностей при этом равны:

X(1)=51; X(2)=64; X(3)=48; X(4)=91; X(5)=108;

X(6) = 105; X(7) = 71; X(8) = 56; X(9) = 38, и выбранные пикселы представляются и запоминаются в вычислителе 9 в последовательности (см. табл. 4).

Далее уровни сигналов яркости в рассчитанной пиксельной последовательности запоминаются в блоке памяти 10, а их координаты (x, y) - через сумматор 11 поступают на вход управления блока памяти опорного кадра 6, обеспечивая запись в блоке 12 памяти уровней пикселов сравнения уровней пикселов с координатами

RU 2137194 C1

RU 2137194 C1

$(x-V_{OX}-V_{x,y}-V_{OY}-V_Y)$,

где (V_{OX}, V_{OY})- координаты начального вектора смещения, которые определяются, например, из результатов оценки движения соответствующих макроблоков в предыдущем кадре или из иных соображений,

(V_X, V_Y) - координаты текущего вектора смещения макроблока, при изменении которых производится анализ контрольной суммы разностей модулей уровней выбранных пикселов в макроблоке и пикселов в опорном кадре.

С блоков памяти 10 и 12 в указанной выше последовательности значения соответствующих уровней пикселов поступают на блок попиксельного высчитания 13 и с его выхода - на блок суммирования модулей 14. Блоки 13 и 14 совместно реализуют операцию, определяемую соотношением (2).

Последовательно вычисляемая в блоке 14 сумма поступает параллельно на блок сравнения сумм 15 и блок вычисления 16 минимальной суммы и соответствующих ей векторов движения. Первоначально при переходе от одного макроблока к другому в блоке 16 запоминается рассчитанный ранее начальный вектор движения (V_{OX}, V_{OY}), а сумма устанавливается нереально большой величиной.

При вычислении суммы для начального вектора движения эта сумма запоминается в блоке 16 и становится опорной для последующих расчетов. Если эта сумма при первом же расчете оказывается ниже некоторой установленной малой величины, то считается, что вектор движения найден. В противном случае по окончании расчета суммы при нулевом векторе (V_X, V_Y) блок сравнения 15 выдает команду на изменение координат вектора движения в счетчик перемещений 17 и обнуление результатов суммирования в блок 14. Обычно для обеспечения эффективного анализа вектора движения перемещение текущего вектора движения осуществляется от точки (V_{OX}, V_{OY}) по расходящейся спирали в пределах заданной зоны анализа.

В счетчике 17 формируется вектор $(-V_{OX} - V_X, - V_{OY} - V_Y)$, который затем суммируется в схеме 11 с текущими значениями координат выбранных характерных пикселов макроблока. Полученные координаты определяют запись уровней яркости соответствующих пикселов из блока 6 в блок 12, и процесс анализа вектора движения продолжается до тех пор, пока не найден будет истинный вектор движения макроблока (когда минимальная сумма, вычисляемая блоком 16, не станет меньше некоторой установленной минимальной величины), либо пока не будет проанализирована вся заданная зона анализа. Последний случай означает, что подобного макроблока в опорном кадре не существует.

Ускорение процесса анализа движения в соответствии с п. 2 настоящего изобретения определяется непрерывным сравнением текущей суммы и минимальной суммы в блоке 15. Если текущая сумма (до окончания процесса суммирования по всем выбранным пикселам) превышает ранее найденную минимальную сумму, то процесс суммирования прекращается и счетчик 17 перемещений изменяет координаты текущего вектора движения макроблока.

Результаты вычисления векторов движения, определяемые в блоке 16, запоминаются блоком памяти 18 векторов движения из которого результаты расчетов векторов движения, например в предыдущих кадрах (V_{OX}, V_{OY}), используются блоками 16 и 17 в качестве исходных векторов. С блока 18 координаты векторов движения подаются на цифровые выхода устройства 19.

Выполнение операций различных блоков устройства в описанной выше последовательности задается блоком управления, синхронизируемым импульсами с выхода блока синхронизации.

Источники информации

1. K. R.Rao, J.J. Hwang, "Techniques and Standards for Image, Video and Audio Coding", 1996, Prentice-Hall PTR, ISBN 0-13-309907-5.

Формула изобретения:

1. Способ анализа векторов движения деталей в динамических изображениях, включающий преобразование последовательности кадров изображений в цифровую форму, запоминание дискретных отсчетов яркости текущего и опорного кадров, разбиение текущего кадра на макроблоки и поиск вектора движения каждого из макроблоков текущего кадра относительно опорного кадра посредством минимизации по рассматриваемому множеству векторов движения контрольной суммы данного макроблока, являющейся суммой норм попиксельной разности уровней в текущем и опорных кадрах, отличающийся тем, что все множество пикселов рассматриваемого макроблока разбивают на участки, в каждом из которых определенным образом выбирают только один пиксел, и упомянутую контрольную сумму вычисляют лишь по выбранным упомянутым пикселам, при этом упомянутые пикселы в каждом из участков выбирают таким образом, что их уровни в смежных участках наиболее отличаются друг от друга.

2. Способ анализа векторов движения по п. 1, отличающийся тем, что для каждого текущего рассматриваемого значения вектора движения вычисление контрольной суммы по выбранным упомянутым пикселам производят в порядке убывания отклонения значения сигнала в каждом пикселе от среднего значения по всей совокупности упомянутых пикселов и прерывают дальнейшее вычисление контрольной суммы в случае, когда она превышает значение минимальной контрольной суммы, найденной среди всех уже рассмотренных векторов движения.

RU 2137194 C1

RU 2137194 C1

Таблица 1

№ участка	Координаты пиксела		Уровень сигнала яркости
	x	*y*	
1	5	5	164
2	8	1	132
3	15	1	151
4	5	10	87
5	6	11	24
6	15	8	178
7	2	15	236
8	6	13	124
9	13	16	112

Таблица 2

№ участка	Координаты пиксела		Уровень сигнала яркости
	x	*y*	
1	2	1	81
2	11	1	196
3	14	3	84
4	1	9	221
5	6	11	24
6	12	9	237
7	3	12	61
8	8	13	188
9	16	16	94

Таблица 3

№ участка	Координаты пиксела		Уровень сигнала яркости	Модуль разности
	x	*y*		
5	6	11	24	110

RU 2137194 C1

RU 2137194 C1

Продолжение таблицы 3

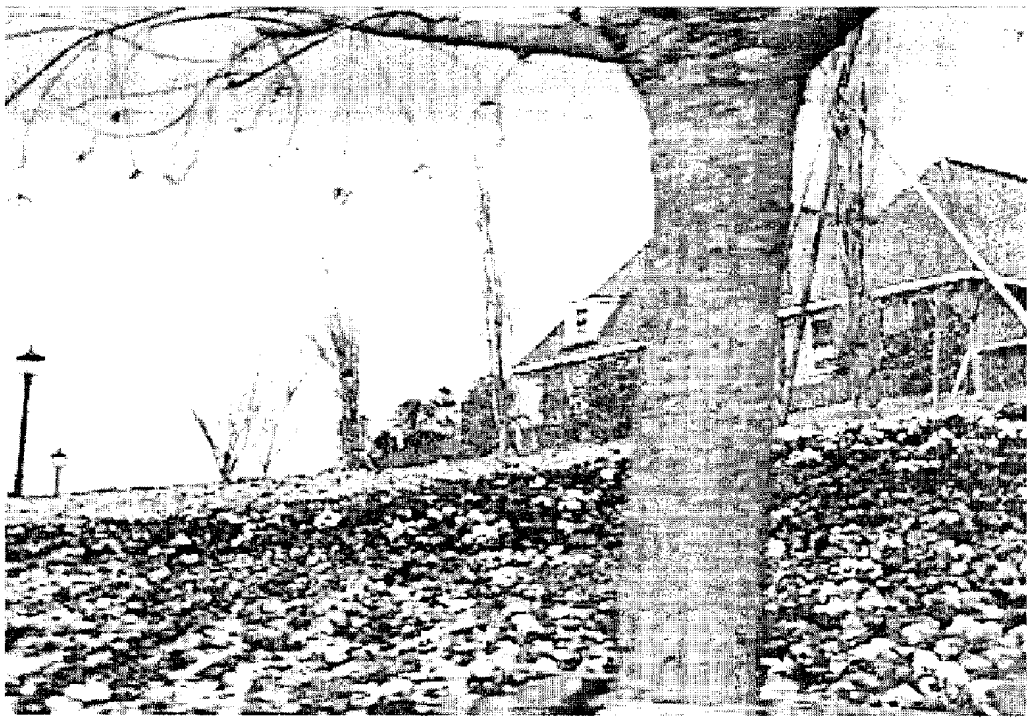
7	2	15	236	102
4	5	10	87	47
6	15	8	178	44
1	5	5	164	30
9	13	16	112	22
3	15	1	151	17
8	6	13	124	10
2	8	1	132	2

Таблица 4

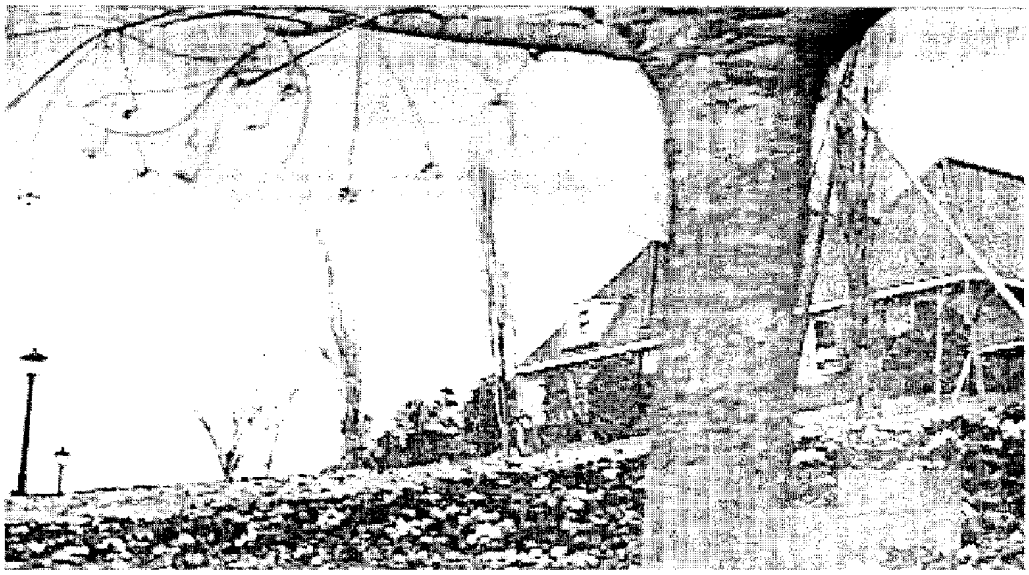
№ участка	Координаты пиксела		Уровень сигнала яркости	Модуль разности
	x	*y*		
5	6	11	24	108
6	12	9	237	105
4	1	9	221	91
7	3	12	61	71
2	11	1	196	64
8	8	13	188	56
1	2	1	81	51
3	14	3	84	48
9	16	16	94	38

RU 2137194 C1

RU 2137194 C1

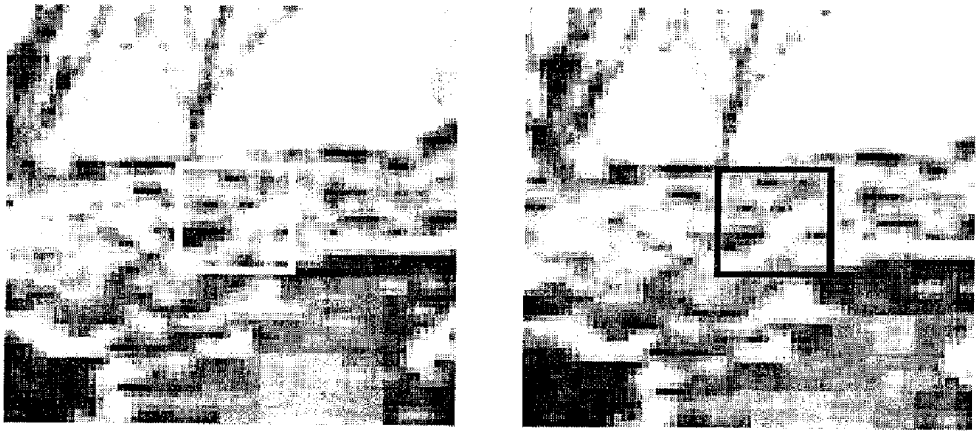


(a)



RU 2137194 C1

RU 2137194 C1

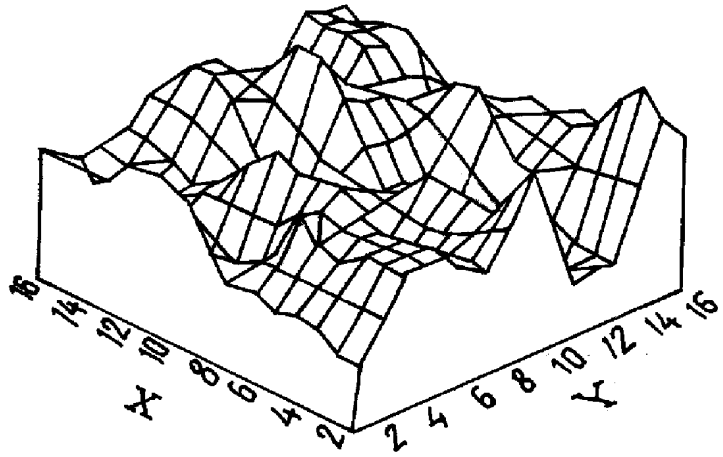


(а) (б)

Фиг.3.

RU 2137194 C1

RU 2137194 C1



а

Таблица уровней пикселов в макроблоке, разделенном на 9 участков

y \ x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	Участок № 1					Участок № 2						Участок № 3				
1	82	81	97	102	94	105	100	132	171	189	196	181	151	140	151	151
2	126	125	127	125	127	127	112	104	117	137	150	144	125	117	125	135
3	167	165	168	142	153	174	115	62	64	87	118	121	86	84	106	117
4	166	163	158	154	161	165	147	126	123	127	125	114	101	103	115	128
5	166	160	156	156	164	183	178	174	186	168	138	115	100	119	131	131
	Участок № 4					Участок № 5						Участок № 6				
6	140	140	135	128	125	132	145	153	147	128	117	126	143	157	159	157
7	120	122	116	104	75	80	117	136	123	80	82	139	186	203	195	170
8	164	149	131	120	115	115	120	126	134	143	160	182	191	187	178	170
9	221	179	136	142	147	151	125	111	152	197	235	237	204	172	166	163
10	142	120	100	90	87	85	88	109	148	182	195	188	170	150	132	125
11	70	61	55	39	23	24	41	91	151	180	171	141	139	126	95	78
	Участок № 7					Участок № 8						Участок № 9				
12	78	63	61	64	69	80	103	136	159	163	153	148	153	151	139	131
13	68	61	68	85	120	124	161	188	179	157	134	143	178	186	177	180
14	146	143	138	134	140	148	158	162	151	133	124	140	166	174	170	166
15	210	236	210	169	172	161	163	148	124	110	97	127	174	169	155	161
16	181	189	176	145	120	113	121	124	112	94	86	97	112	110	99	94

(б)

Фиг. 4.

RU 2137194 C1

RU 2137194 C1

Уровень пиксела	23	[24]	39	41	55	61	61	61	62	63	64	64	68	68	69	70
Номер участка	4	[5]	4	5	4	7	7	4	2	7	2	7	7	7	7	4

75	78	78	80	80	80	81	82	82	84	85	85	86	86	[87]	87
4	7	6	5	5	8	1	1	5	3	7	5	3	8	[4]	2

88	90	91	94	94	94	95	97	97	97	99	100	100	100	101	102
5	4	5	1	8	9	6	1	9	8	9	3	4	2	3	1

103	103	104	104	105	106	109	110	110	111	112	[112]	112	113	114	115
3	8	4	2	2	3	5	8	9	5	8	[9]	2	8	3	4

115	115	115	115	116	117	117	117	117	117	118	119	120	120	120	120
2	5	3	3	4	3	5	3	5	2	2	3	4	4	7	5

120	120	121	121	122	123	123	124	[124]	124	124	125	125	125	125	125
7	4	3	8	4	2	5	8	[8]	8	8	4	6	5	1	3

125	125	125	126	126	126	126	126	127	127	127	127	127	128	128	128
1	3	2	1	6	2	5	6	9	2	1	2	1	3	4	5

131	131	131	131	132	{132}	[132]	133	134	134	134	135	135	136	136	136
3	4	9	3	5	{6}	[2]	8	7	5	8	3	4	8	4	5

137	138	138	139	139	139	140	140	140	140	140	141	142	142	142	143
2	2	7	6	6	9	3	4	4	7	9	6	4	4	1	6

143	143	143	144	145	145	146	147	147	147	148	148	148	148	149	150
7	5	9	3	5	7	7	5	2	4	5	8	9	8	4	6

150	151	151	151	151	151	{151}	[151]	152	153	153	153	153	154	155	156
2	3	3	8	5	9	{5}	[3]	5	5	1	8	9	1	9	1

156	157	157	157	158	158	159	159	160	160	161	161	161	161	162	163
1	8	6	6	1	8	8	6	5	1	8	8	1	9	8	1

163	163	163	[164]	164	165	165	166	166	166	166	166	167	168	168	169
8	6	8	[1]	4	2	1	6	9	9	1	1	1	1	2	9

169	170	170	170	170	171	171	172	172	174	174	174	174	176	177	178
7	6	6	9	6	2	5	7	6	9	2	2	9	7	9	9

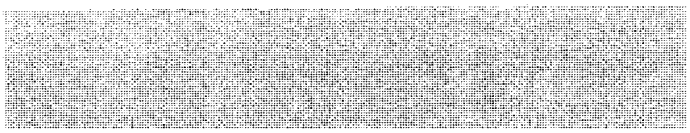
178	[178]	179	179	180	180	181	181	182	182	183	186	186	186	187	188
2	[6]	4	8	9	5	3	7	6	5	2	6	2	9	6	6

188	189	189	191	195	195	196	197	203	204	210	210	221	235	[236]	237
8	7	2	6	5	6	2	5	6	6	7	7	4	5	[7]	6

Фиг. 5.

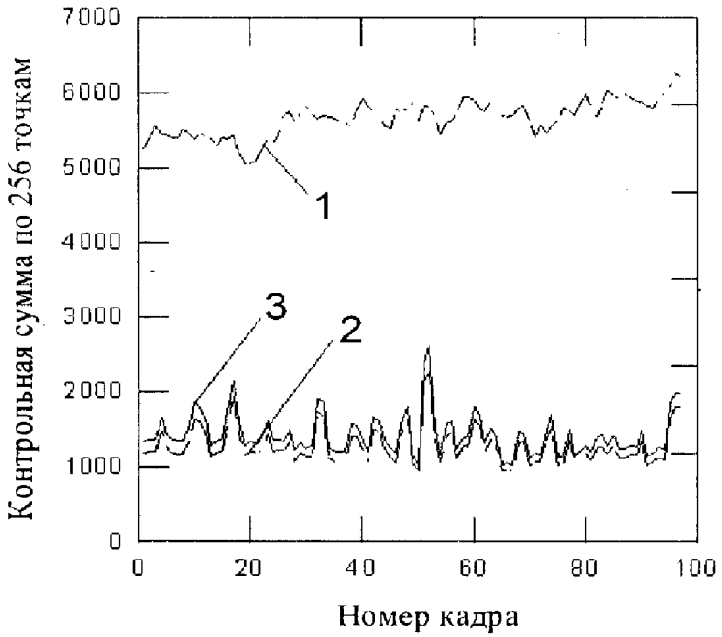
RU 2137194 C1

RU 2137194 C1



RU 2137194 C1

RU 2137194 C1



Фиг. 8.

RU 2137194 C1

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,934	01/25/2006	Miska Hannuksela	TPO.004.C2	6610

76385 7590 05/25/2010
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

MAIL DATE	DELIVERY MODE
05/25/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/338,934	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	SIU M. LEE	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 January 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 January 2006* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☒ Certified copies of the priority documents have been received in Application No. 10/306942.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *See Continuation Sheet*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :6/19/2006, 8/18/2006, 11/24/2006, 10/22/2007, 6/16/2008, 3/12/2009, 4/20/2009, 2/08/2010, 2/18/2010, 3/30/2010.

DETAILED ACTION

Claim Objections

1. Claims 1-4, 5-8, and 10-12 are objected to because of the following informalities:

Claim 1, line 4 recites “the previous reference image frame”; there is a lack of antecedent basis for this limitation.

Claims 5, line 4 recites “the previous reference image frame”; there is a lack of antecedent basis for this limitation.

Claim 10, line 3 recites “the previous reference image frame”; there is a lack of antecedent basis for this limitation.

Appropriate correction is required.

Drawings

2. The drawings are objected to because Figure 1 does not have sufficiently descriptive labels. Blank boxes in drawings should be labeled descriptively unless it is a well-known component. For example, the blank box, "100" should be labeled as "Multimedia Source". Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as “amended.” If a drawing figure is to be canceled, the appropriate figure must be removed from the

replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 5-8 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claimed "video encoder" is non-structural per se and because the specification discloses a "software" embodiment (page 5, paragraph 0014 and page 33, paragraph 0097). Therefore, a reasonable interpretation in light of the specification leads to the conclusion that the claim video encoder as pure software (further confirm in claim 9 that computer program code for encoding a compressed video sequence, claim 9, line 7), which does not fall within the definition of a process, machine, manufacture or composition of matter.

Claim 13 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claimed "video decoder" is non-structural per se and because the specification discloses a "software" embodiment (page 5, paragraph 0014 and page 33, paragraph 0097). Therefore, a broadest reasonable interpretation in light of the specification leads to the conclusion that the claim video decoder as pure software (further confirm in claim 9 that computer program code for decoding from the video sequence, claim 14, line 7), which does not fall within the definition of a process, machine, manufacture or composition of matter.

Claim 9 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claim is drawn to a "computer program product stored in a computer readable medium". The specification does not define the computer readable medium, therefore, it includes transitory and non-transitory computer readable medium. The broadest reasonable interpretation of the claim in light of the specification concludes that the claim as a whole covers a transitory signal, which does not fall within the definition of a process, machine, manufacture, or composition of matter (In re Nuijten). Therefore, claim 9 does not fall within a statutory category (the rejection can be overcome by specifying the computer readable medium as "non-transitory computer readable medium").

Claim 14 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine,

manufacture or composition of matter. The claim is drawn to a "computer program product stored in a computer readable medium". The specification does not define the computer readable medium, therefore, it includes transitory and non-transitory computer readable medium. The broadest reasonable interpretation of the claim in light of the specification concludes that the claim as a whole covers a transitory signal, which does not fall within the definition of a process, machine, manufacture, or composition of matter (In re Nuijten). Therefore, claim 9 does not fall within a statutory category (the rejection can be overcome by specifying the computer readable medium as "non-transitory computer readable medium").

Claim 15-16 are directed to a "video signal"; a signal is not limited to that which falls within a statutory category of invention because it is not limit to a process, machine, manufacture, or a composition of matter. Instead, it includes a form of energy. Energy does not fall with in a statutory category since it is clearly not a series of steps or acts to constitute a process, not a mechanical device or combination of mechanical device to constitute a machine, not a tangible physical article or object which is some form of matter to be a product and constitute a manufacture, and not a composite of two or more substances to constitute a composite of matter.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 5-8 are rejected under 112, 1st paragraph as being a single component (video encoder) claim, i.e., where a single component (video encoder) recitation does not appear in combination with another recited component, is subject to an undue breadth rejection. The single component which covered every conceivable ways for achieving the stated purpose was held nonenabling for the scope of claim because the specification disclosed at most only those known to the inventor. See MPEP §2164.08(a).

Claim 13 is rejected under 112, 1st paragraph as being a single component (video decoder) claim, i.e., where a single component (video decoder) recitation does not appear in combination with another recited component, is subject to an undue breadth rejection. The single component which covered every conceivable ways for achieving the stated purpose was held nonenabling for the scope of claim because the specification disclosed at most only those known to the inventor. See MPEP §2164.08(a).

Double Patenting

5. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422

F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

6. Claims 1-2, and 4 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 9 of copending Application No. 10/306942 in view of Kim et al. (US 2003/0063806 A1, hereinafter Kim).

Claim	Instant application	Claim	Co-pending application 10/306942
1	A method for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order, the	9	Claim 1 A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, **the video frames of the first frame format being independent of the**

	method comprising: encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence.		**other video frames**, and **the video frames of the second frame format being predicted from at least one of the other video frames**, the method comprising encoding, at an encoder, the video sequence as at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format; and encoding, at the encoder, the video sequence as at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and **at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence**; and encoding, at the
2	A method according to claim 1, further comprising: encoding the indication into the video sequence as a separate flag included in the header of a slice.		
4	A method according to claim 1, further comprising: encoding into the video sequence an identifier value for the independent sequence.		

			encoder, into the video sequence information indicating which video frames belong to the second sub-sequence. Claim 3 A method according to claim 2, further comprising coding the video sequence into a plural number of scalability layers; and determining the dependency of the video frames of the second sub-sequence **such that at least one video frame of the second sub-sequence is predicted from a group comprising a video frame of a higher scalability layer, a video frame of another sub-sequence in the same scalability layer**. Claim 7 A method according to claim 3, further comprising **determining a**

			unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number. Claim 9 A method according to claim 7, further comprising **adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence**.

This is a provisional obviousness-type double patenting rejection.

(1) Regarding claim 1:

The co-pending application discloses a method for encoding a video sequence comprising an independent sequence of image frames (video sequence comprising video frame coded according to at least a first and second fame format), wherein at least one reference image frame (second sub-sequence) is predictable from at least one previous image frame that is earlier than a previous reference image frame in decoding order (at least one video frame of the second sub-sequence has been

predicted from at least one video frame of the first sub-sequence or a video frame of another sub-sequence in the same scalability layer), the method comprising:

encoding into the video sequence an indication (identifier) of at least one image frame (determined a unique identifier for each video frame), which is the first image frame, in decoding order, of the independent sequence (since **each** frame has an identifier added to the header field of the video sequence, therefore, it included the first image frame), said at least one reference image frame being included in the sequence (the video sequence includes at least a second sub-sequence, therefore, the claimed at least one reference image frame is included in the video sequence).

Claim 9 of the instant application does not explicitly disclose at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order.

However, Kim discloses an independent video sequence in figure 7 that at least one reference image frame P frame 708) is predictable from at least one previous image frame (I frame 704) that is earlier than a previous reference image frame in decoding order (P frame 706).

It is desirable to have at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order because it reduces latency when decoding the video sequence. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the teaching of Kim in the method of the co-pending application to reduce latency when decoding the video sequence.

(2) Regarding claim 2 (the examiner interprets "a slice" as the video sequence as claim in claim 1):

The copending application discloses:

encoding the indication into the video sequence as a separate flag included in the header of a slice (adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence, the examiner interprets the adding process as the encoding process as claim).

(3) Regarding claim 4:

The co-pending application discloses:

encoding into the video sequence an identifier value for the independent sequence (determining a unique identifier for each video frame and adding the identifier to the header field of the video sequence, the examiner interpret adding the unique identifier to the header as encoding the identifier into the video sequence).

7. Claim 3 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 9 of copending Application No. 10/306942 in view of Kim et al. (US 2003/0063806 A1, hereinafter Kim) as apply to claim 1 and further in view of Kawashima et al. (US 4,390,966).

Copending application and Kim discloses all subject matter as discussed in claim 1 and copending application further discloses the encoding identifier values for image frames according to a numbering scheme (layer number, sub-sequence identifier and

image number), but fail to disclose resetting the identifier value for the indicated first image frame of an independent sequence.

However, Kawashima discloses a controller that setting or resetting flags as desired according to user instruction (column 6, lines 37-40).

It is desirable to reset the identifier value for the indicated first image frame of an independent sequence because since the video sequence is independent of other sequences, it would simplify the identifier value and reset the identifier value. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the resetting step as taught by Kawashima in the method of copending application and Kim to simplify the identifier value of the video sequence.

Kim et al. (US 2003/0063806 A1, hereinafter Kim) in view of Ueda (US 6,094,456)

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9. Claims 1 and 5 are rejected under 35 U.S.C. 102(e) as being anticipated by Kim et al. (US 2003/0063806 A1, hereinafter Kim).

Kim discloses a method and encoder (encoder discloses in figure 1B and 1C) for encoding a video sequence comprising an independent sequence of image frames (independent sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising:

encoding into the video sequence an indication of at least one image frame (a video coding apparatus is provided to receive an indication that a first frame is to be intracoded, paragraph 0018, the examiner interprets the first intracoded frame as the claimed indication), which is the first image frame (the first frame is intracoded), in decoding order, of the independent sequence (independent sequence as shown in figure 7), said at least one reference image frame being included in the sequence (the P frame 708 is included in the sequence).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 1, 2, 4, 6, 8, 10, 11, 13, and 15-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kim et al. (US 2003/0063806 A1, hereinafter Kim) in view of Ueda (US 6,094,456).

(1) Regarding claim 1:

Kim discloses a method and encoder (encoder discloses in figure 1B and 1C) for encoding a video sequence comprising an independent sequence of image frames (independent sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising:

encoding into the video sequence at least one image frame (a video coding apparatus is provided to receive an indication that a first frame is to be intracoded, paragraph 0018, the examiner interprets the first intracoded frame as the claimed indication), which is the first image frame (the first frame is intracoded), in decoding order, of the independent sequence (independent sequence as shown in figure 7), said at least one reference image frame being included in the sequence (the P frame 708 is included in the sequence).

Kim fails to explicitly disclose an indication of the first I frame.

However, Ueda discloses an encoded video sequence that comprises a fixed parameter identifier in the header of an input video code that indicates the type of picture from the picture header (I frame, column 10, lines 38-42).

Since Kim discloses the first frame is an I frame, and Ueda discloses having a fixed parameter identifier in the header to indicate whether the frame is an I frame, it is obvious to employ the teaching of Ueda in the encoding method of Kim to include in the header a fixed parameter identifier for identifying the first I frame in order to improve the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the method of Kim to improve the reproduction speed of the encoded video sequence.

(2) Regarding claims 2 and 6:

Kim discloses all subject matter as discussed in claims 1 and 5 but fail to disclose encoding the indication into the video sequence as a separate flag included in the header of a slice.

However, Ueda discloses an encoded video sequence that comprises a fixed parameter identifier in the header of an input video code that indicates the type of picture from the picture header (column 10, lines 38-42).

Since Kim discloses the first frame is an I frame, and Ueda discloses having a fixed parameter identifier in the header to indicate whether the frame is an I frame, it is obvious to employ the teaching of Ueda in the encoding method of Kim to include in the header a fixed parameter identifier for identifying the first I frame in order to improve the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the method of Kim to improve the reproduction speed of the encoded video sequence.

(3) Regarding claims 4 and 8:

Ueda further discloses encoding into the video sequence an identifier value for the independent sequence (an identifier of six bytes indicative of type of picture, column 7, lines 60-64 and column 10, lines 38-42).

(4) Regarding claims 10 and 13:

Kim discloses a method and decoder for decoding a compressed video sequence (sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising

decoding the first two frames of a sequence that yield the exact frame rate, the first two frame as I frame of figure 7 (paragraph 0071) which is the first image frame (I frame 702 in figure 7 is the first frame of the sequence), in decoding order, of an independent sequence, said sequence including the at least one reference image frame (P frame 708 is included in the sequence) and starting the decoding of the video sequence from said first image frame of the independent sequence (as shown in figure 7, the first I frame 702 and the second I frame 704 is for generating the P frame 706-

712, therefore, the decoding processing is start from I frame 702), whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame (as shown in figure 7, the sequence is independent and I frame 702 is the first frame of the sequence and no other frame prior to frame 702 is required).

Kim does not explicitly disclose decoding from the video sequence an indication of at least one image frame which is the first image frame, in decoding order of an independent sequence.

However, Ueda discloses a method that determines from an identifier in the header and determine if the frame is an I frame (step 111 and 112 of figure 12, column 14, lines 31-35).

It is desirable to decoding from the video sequence an indication of at least one image frame which is the first image frame, in decoding order of an independent sequence because it improves the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the decoding method of Kim to improve the reproduction speed of the encoded video sequence.

(5) Regarding claim 11:

Ueda further discloses wherein the indication is a separate flag in the header of a slice (an identifier of six bytes indicative of type of picture, column 7, lines 60-64 and column 10, lines 38-42).

(6) Regarding claim 15:

Kim discloses a signal comprising a compressed video sequence including an independent sequence of image frames (sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising:

at least one image frame (the first frame is an intracoded frame), which is the first image frame, in decoding order, of the independent sequence (the first I frame 702 is the first frame of the sequence as shown in figure 7), said at least one reference image frame being included in the sequence (P frame 708 is included in the sequence as shown in figure 7).

Kim fails to explicitly disclose an indication of the first I frame.

However, Ueda discloses an encoded video sequence that comprises a fixed parameter identifier in the header of an input video code that indicates the type of picture from the picture header (I frame, column 10, lines 38-42).

Since Kim discloses the first frame is an I frame, and Ueda discloses having a fixed parameter identifier in the header to indicate whether the frame is an I frame, it is obvious to employ the teaching of Ueda in the encoding method of Kim to include in the header a fixed parameter identifier for identifying the first I frame in order to improve the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the method of Kim to improve the reproduction speed of the encoded video sequence.

(7) Regarding claim 16:

Ueda further discloses encoding into the video sequence an identifier value for the independent sequence (an identifier of six bytes indicative of type of picture, column 7, lines 60-64 and column 10, lines 38-42).

12. Claims 3 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kim et al. (US 2003/0063806 A1, hereinafter Kim) in view of Ueda (US 6,094,456) and Kawashima et al. (US 4,390,966).

Kim discloses all subject matter as discussed in claims 1 and 5 except (a) encoding identifier values for image frames according to a numbering scheme, and (b) resetting the identifier value for the indicated first image frame of an independent sequence.

With respect to (a), Ueda discloses a fixed parameter identifier of six bits in the header of an input video code that indicates the type of picture from the picture header (column 7, lines 60-64 and column 10, lines 38-42, the examiner interprets the bits as a representation of numbering scheme in binary).

Since Kim discloses the first frame is an I frame, and Ueda discloses having a fixed parameter identifier in the header to indicate whether the frame is an I frame, it is

obvious to employ the teaching of Ueda in the encoding method of Kim to include in the header a fixed parameter identifier for identifying the first I frame in order to improve the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the method of Kim to improve the reproduction speed of the encoded video sequence.

With respect to (b), Kawashima discloses a controller that setting or resetting flags as desired according to user instruction (column 6, lines 37-40).

It is desirable to reset the identifier value for the indicated first image frame of an independent sequence because since the video sequence is independent of other sequences, it would simplify the identifier value and reset the identifier value. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the resetting step as taught by Kawashima in the method of copending application and Kim to simplify the identifier value of the video sequence.

13. Claims 9 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kim et al. (US 2003/0063806 A1, hereinafter Kim) in view of Langberg et al. (US 5,852,630).

(1) Regarding claim 9:

Kim discloses a method and encoder (encoder discloses in figure 1B and 1C) for encoding a video sequence comprising an independent sequence of image frames

(independent sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising:

encoding into the video sequence an indication of at least one image frame (a video coding apparatus is provided to receive an indication that a first frame is to be intracoded, paragraph 0018, the examiner interprets the first intracoded frame as the claimed indication), which is the first image frame (the first frame is intracoded), in decoding order, of the independent sequence (independent sequence as shown in figure 7), said at least one reference image frame being included in the sequence (the P frame 708 is included in the sequence).

Kim fails to disclose implementing the encoder by a computer program product, stored on a computer readable medium and executable in a data processing unit.

However, Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Kim would have been implemented in a software. The implemented software would perform same function of the hardware for less expense, adaptability,

and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg to implement the method and encoder of Kim in order to reduce cost and improve the adaptability and flexibility of the communication system.

(2) Regarding claim 14:

Kim discloses a method and decoder for decoding a compressed video sequence (sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising

decoding the first two frames of a sequence that yield the exact frame rate, the first two frame as I frame of figure 7 (paragraph 0071) which is the first image frame (I frame 702 in figure 7 is the first frame of the sequence), in decoding order, of an independent sequence, said sequence including the at least one reference image frame (P frame 708 is included in the sequence) and starting the decoding of the video sequence from said first image frame of the independent sequence (as shown in figure 7, the first I frame 702 and the second I frame 704 is for generating the P frame 706-712, therefore, the decoding processing is start from I frame 702), whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame (as shown in figure 7, the sequence is independent and I frame 702 is the first frame of the sequence and no other frame prior to frame 702 is required).

Kim does not explicitly disclose (a) decoding from the video sequence an indication of at least one image frame which is the first image frame, in decoding order of an independent sequence, and (b) implementing the method by a computer program product, stored on a computer readable medium and executable in a data processing unit.

With respect to (a), Ueda discloses a method that determines from an identifier in the header and determine if the frame is a I frame (step 111 and 112 of figure 12, column 14, lines 31-35).

It is desirable to decoding from the video sequence an indication of at least one image frame which is the first image frame, in decoding order of an independent sequence because it improves the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the decoding method of Kim to improve the reproduction speed of the encoded video sequence.

With respect to (b), Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Kim would have been implemented in a software. The implemented

software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg to implement the method and encoder of Kim and Ueda in order to reduce cost and improve the adaptability and flexibility of the communication system.

14. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kim et al. (US 2003/0063806 A1, hereinafter Kim) in view of Ueda (US 6,094,456) as applied to claim 10 and further in vie of Kawashima et al. (US 4,390,966).

Kim and Ueda disclose all subject matter as discussed in claim 10, and Ueda further discloses a fixed parameter identifier of six bits in the header of an input video code that indicates the type of picture from the picture header (column 7, lines 60-64 and column 10, lines 38-42, the examiner interprets the bits as a representation of numbering scheme in binary).

Kim and Ueda fail to disclose resetting the identifier value for the indicated first image frame of an independent sequence.

However, Kawashima discloses a controller that setting or resetting flags as desired according to user instruction (column 6, lines 37-40).

It is desirable to reset the identifier value for the indicated first image frame of an independent sequence because since the video sequence is independent of other sequences, it would simplify the identifier value and reset the identifier value. Therefore, it would have been obvious to one of ordinary skill in the art at the time of

invention to employ the resetting step as taught by Kawashima in the method of copending application and Kim to simplify the identifier value of the video sequence.

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Cuccia (US 202/0051621 A1) discloses an instant replay on digital video optimized using non MPEG frame tag.

Hannuksela et al. (US 2001/0040700 A1) discloses a video coding.

Apostolopoulos (US 7,103,669 B2) discloses a video communication method and system employing multiple state encoding and path diversity.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU M. LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 7:30-4:00 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Examiner, Art Unit 2611
5/21/2010

/CHIEH M FAN/

Supervisory Patent Examiner, Art Unit 2611

Electronic Patent Application Fee Transmittal

Application Number:	11338996
Filing Date:	25-Jan-2006
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Filer:	Steven R. Funk/Tracey Dotter
Attorney Docket Number:	TPO.004.C1

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
		Total in USD ($)		**180**

Electronic Acknowledgement Receipt

EFS ID:	8190475
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Steven R. Funk
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	10-AUG-2010
Filing Date:	25-JAN-2006
Time Stamp:	17:31:01
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$180
RAM confirmation Number	3964
Deposit Account	503581
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-10-TPO004C1.pdf	55389 38ad6f68253374c3da720b0497bc7eee9b9147fd	no	2
Warnings:					
Information:					
2	Information Disclosure Statement (IDS) Filed (SB/08)	1449-10-TPO004C1.pdf	53779 9ae450ccc23f415e3eb344ca00e9829ce382ba81	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	FH-TAPA004PA.pdf	999816 e0f3b198e0372dfeee0d292f6afef40141aed93c	no	27
Warnings:					
Information:					
4	NPL Documents	Illgner.pdf	524122 72978d8c3d123a922894913ff66ff592b86674fa	no	16
Warnings:					
Information:					
5	NPL Documents	Informationtechnology.pdf	1641793 d76b603427c9443ca71ff18667af1f1834b69a97	no	174
Warnings:					
Information:					
6	NPL Documents	RUAllowance.pdf	4724947 318f5d65aaeeedf8f2b368397e91b944baced92a	no	11
Warnings:					
Information:					
7	Foreign Reference	RU2137194C1.pdf	973318 31ad8c486895acf06c549d3f83a00eff14e65d24	no	15
Warnings:					
Information:					
8	NPL Documents	TPO004C2-OA05252010.pdf	1003692 e2a3f23f274433505e5b4fa80643d76d83254e99	no	29
Warnings:					
Information:					

9	Fee Worksheet (PTO-875)	fee-info.pdf	30212 bc9f912c355b965e90451f8e2be424dad862ad9a	no	2

Warnings:	
Information:	
Total Files Size (in bytes):	10007068

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. §1.97(c)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant respectfully submits the items of information on the enclosed Form 1449 for the attention of the Examiner in the above-identified application.

This statement should be considered because it is submitted after the mailing date of the first Office Action on the merits but before the mailing date of a final Office Action under 37 C.F.R. §1.113 or a Notice of Allowance under 37 C.F.R. §1.311. Please charge Deposit Account 50-3581 (TPO.004.C1) the amount of $180.00 under 37 C.F.R. §1.17(p) for consideration of the item.

In accordance with 37 C.F.R. §1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that the reference is "prior art" within the meaning of 35 U.S.C. §§102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. §1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Steven R. Funk/
Steven R. Funk
Reg. No. 37,830

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on September 10, 2010.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action mailed June 17, 2010, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.111.

Claims 1-25 remain pending in the application. Claims 1-3, 5-12, 14-20, and 22-25 have been amended. No claims have been added, and claims 26-31 have been canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 8.

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions of claims in the application.

1. (Currently amended) A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising:

decoding from the compressed video sequence an indication related to at least one discontinuity of numbering of the image frames;

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to [[a]]the at least one discontinuity in the numbering of the image frames; and

using the image frames in the buffer memory in the decoding process.

2. (Currently amended) A method according to claim 1, wherein said indication informs that the at least one discontinuity in the numbering of the image frames in the video sequence [[are]]is intentional.

3. (Currently amended) A method according to claim 1, further comprising

using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames.

4. (Original) A method according to claim 1, wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.

5. (Currently amended) A method according to claim 1, further comprising

continuing the entering of the image frames comprised by the compressed video sequence ~~in question~~ into the buffer memory from [[the]]a correct image frame number after the buffer memory has been configured to provide the number of image frames

corresponding to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames.

6. (Currently amended) A method according to claim 1, further comprising

entering into the buffer memory a number of filler frames corresponding to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames.

7. (Currently amended) A method according to claim 6, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the ~~actual~~compressed video sequence.

8. (Currently amended) A method according to claim 1, wherein

~~the step of~~ configuring the buffer memory to provide a number of image frames corresponding to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.

9. (Currently amended) A method according to claim 3, further comprising

detecting an error in the compressed video sequence on [[the]]a basis of a reference in the decoding process to an image frame corresponding to [[a]]the at least one discontinuity in the numbering of image frames.

10. (Currently amended) A video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding, the video decoder being ~~arranged~~configured to

decode from the decompressed video sequence an indication related to at least one discontinuity of numbering of the image frames;

configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to [[a]]the at least one discontinuity in the numbering of the image frames; and

use the image frames in the buffer memory in the decoding process.

11. (Currently amended) A video decoder according to claim 10, wherein

said indication informs that ~~discontinuities~~the at least one discontinuity in the numbering of the image frames in the video sequence [[are]]is intentional.

12. (Currently amended) A video decoder according to claim 10, wherein the video decoder is further ~~arranged~~configured to

use the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames.

13. (Original) A video decoder according to claim 10, wherein

the number of image frames in the buffer memory are used in sliding window buffering of image frames.

14. (Currently amended) A video decoder according to claim 10, wherein the video decoder is further ~~arranged~~configured to

continue the entering of the image frames comprised by the compressed video sequence ~~in question~~ into the buffer memory from [[the]]a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames.

15. (Currently amended) A video decoder according to claim 10, wherein the video decoder is further ~~arranged~~configured to

enter into the buffer memory a number of filler frames corresponding to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames.

16. (Currently amended) A video decoder according to claim 15, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the ~~actual~~compressed video sequence.

17. (Currently amended) A video decoder according to claim 10, wherein the video decoder is further ~~arranged~~configured to

configure the buffer memory to provide a number of image frames corresponding to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

18. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, image frames of said video sequence being entered into a buffer memory in connection with the decoding, the computer program product comprising:

a computer program code for decoding from the compressed video sequence an indication related to at least one discontinuity of numbering of the image frames;

a computer program code for configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to [[a]]the at least one discontinuity in the numbering of the image frames; and

a computer program code for including the image frames in the buffer memory in the decoding process.

19. (Currently amended) A computer program product according to claim 18, wherein said indication informs that ~~discontinuities~~the at least one discontinuity in the numbering of the image frames in the video sequence [[are]]is intentional.

20. (Currently amended) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for including the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames.

21. (Original) A computer program product according to claim 18, the computer program product further comprising

a computer program code for using the number of image frames in the buffer memory in sliding window buffering of image frames.

22. (Currently amended) A computer program product according to claim 18, the computer program product further comprising

a computer program code for continuing the entering of the image frames comprised by the compressed video sequence ~~in question~~ into the buffer memory from [[the]]a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames.

23. (Currently amended) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for entering into the buffer memory a number of filler frames corresponding to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames.

24. (Currently amended) A computer program product according to claim 23, the computer program product further comprising:

a computer program code for designating said filler frames by an identifier indicating that the filler frames do not belong to the ~~actual~~compressed video sequence.

25. (Currently amended) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for configuring the buffer memory to provide a number of image frames corresponding to the ~~discontinuities~~at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

26-31. (Canceled)

REMARKS/ARGUMENTS

Applicant appreciates the indication of allowability for claims 7-9. In view of the amendments and remarks herein, these claims are believed to be allowable in their pending format; therefore, Applicant declines the opportunity to rewrite these claims at this time.

The rejections presented in the Office action dated June 17, 2010 (hereinafter Office action) have been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

Claims 26-31 have been canceled, without prejudice, rendering the respective objections to, and rejections of, these claims moot. Applicant accordingly requests that the respective objections to, and rejections of, each of these claims be withdrawn.

With respect to the objection to claims 1, 5, 9, 10, 14, 18, 22, and 24, each of these claims has been amended to provide antecedent basis for the identified terms. In addition, further dependent claims have been amended to provide consistency with the changes to these claims. Applicant accordingly requests that the objection be removed.

With respect to the objection to Fig. 1, Applicant respectfully traverses. In contrast to the assertion that the figure includes "blank boxes", each of the boxes of Fig. 1 is labeled with a reference numeral in accordance with MPEP § 608.02. Moreover, no requirement has been cited for the request to label the boxes with text. Since Fig. 1 complies with the drawing requirements, and in an effort to maintain consistency with the parent application, Applicant requests that the objection be removed.

With respect to the § 101 rejection of claims 10-17, Applicant respectfully traverses. The Office action incorrectly asserts that the claimed video decoder is "pure software". While the specification refers to embodiments involving computer readable media storing program code (*e.g.*, claim 18), the claimed video decoder is a device constituting statutory subject matter. The term "decoder" is well-known in the art to refer to a device or circuit. Notably, the specification illustrates structural components of the decoder (*e.g.*, demultiplexer 116 and decompressor 118), and independent claim 10 recites that the video decoder is connected to a buffer memory. Paragraph [0095] teaches examples of a decoder

including a low bit rate video decoder as described in the ITU-T recommendation H.263 or H.26L. Moreover, paragraph [0096] describes positioning of the decoder that would not be possible if it was "pure software". Contrary to the assertion, the claimed video decoder is a device constituting statutory subject matter. Applicant accordingly requests that the rejection be withdrawn.

With respect to the § 101 rejection of claims 18-25, the claims have been amended to indicate that the claimed computer readable medium is non-transitory in accordance with the Examiner's suggestion. Applicant notes that non-transitory computer readable media comprise all computer readable media, with the sole exception being a transitory, propagating signal. Since an interpretation of the claims is now limited to statutory subject matter, Applicant submits that the rejection has been overcome and accordingly requests that the rejection be withdrawn.

With respect to the provisional rejections of claims 1-6, 10-15, and 18-23 under the judicially-created doctrine of obviousness-type double patenting over co-pending and commonly-owned patent application no. 10/306,942 (hereinafter "the '942 application"), Applicant will consider filing a Terminal Disclaimer once the instant application is in condition for allowance or the '942 application issues as a patent. However, Applicant respectfully asserts that, in view of the amendments to the claims and arguments made above, the Examiner is compelled to withdraw the rejections of the claims. Once withdrawn, the only rejection remaining in the instant application would be the provisional obviousness-type double patenting rejection. In view of MPEP § 804 I(B), Applicant respectfully requests that the provisional obviousness-type double patenting rejection be withdrawn and that the instant application be permitted to issue as a patent.

It should also be noted that Applicant does not acquiesce to the Examiner's statements or conclusions concerning what would have been obvious to one of ordinary skill in the art, obvious design choices, inherent, common knowledge at the time of Applicant's invention, officially noticed facts, and the like. Applicant reserves the right to address in detail the Examiner's characterizations, conclusions, and rejections in future prosecution.

Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C1) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the undersigned attorney of record invites the Examiner to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

Date: September 10, 2010

By: /Erin Nichols Matkaiti/

Erin Nichols Matkaiti
Reg. No. 57,125

Electronic Acknowledgement Receipt

EFS ID:	8395030
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	10-SEP-2010
Filing Date:	25-JAN-2006
Time Stamp:	15:31:25
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		OAR-TPO004C1-OA17062010.pdf	89215 26aba7cc752c3fc6ba6ddd2ef825b135aa56074e	yes	10

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment/Req. Reconsideration-After Non-Final Reject	1	1
	Claims	2	7
	Applicant Arguments/Remarks Made in an Amendment	8	10
Warnings:			
Information:			
	Total Files Size (in bytes):	89215	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
11/338,996	01/25/2006	

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY ☐ RATE ($)	SMALL ENTITY ☐ FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
FOR	NUMBER FILED	NUMBER EXTRA					
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	09/10/2010	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 25	Minus	** 31	= 0	X $ =		OR	X $52=	0
	Independent (37 CFR 1.16(h))	* 2	Minus	***6	= 0	X $ =		OR	X $220=	0
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/MONICA HARRIS/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	9/17/2010	Office Action Response dated 9/17/2010 from U.S. Serial No. 11/338,934, 10 pages.
	9/15/2010	Office Action Response dated 9/15/2010 from U.S. Serial No. 10/306,942, 12 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Diep, N.
Serial No.:	10/306,942	Group Art Unit:	2621
Filed:	November 29, 2002	Docket No.:	TAPA.004PA
Confirmation No.:	3642	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on September 15, 2010.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action mailed June 23, 2010, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.111.

Claims 1, 3-13, 15-24, 26, and 27, remain pending in the application. Claims 1, 3, 15, 22, and 23 have been amended. No claims have been added, and claim 2 has been canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 10.

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions of claims in the application.

1. (Currently amended) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of [[the]]other video frames, and the video frames of the second frame format being predicted from at least one ~~of the~~ other video ~~frames~~frame, the method comprising

encoding, at an encoder, the video sequence as at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format; and

encoding, at the encoder, the video sequence as at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence;

determining a dependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence; and

encoding~~, at the encoder,~~ said dependency into the video sequence ~~information indicating which video frames belong to the second sub-sequence~~.

2. (Canceled)

3. (Currently amended) A method according to claim [[2]]1, further comprising

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from a group comprising

- a video frame of a higher scalability layer,

- a video frame of another sub-sequence in the same scalability layer.

4. (Original) A method according to claim 3, further comprising

determining the dependencies of the video frames of the second sub-sequence on the basis of at least a scalability layer identifier and a sub-sequence identifier.

5. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

6. (Original) A method according to claim 3, further comprising

coding the first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

7. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.

8. (Original) A method according to claim 3, further comprising

determining a unique identifier for each video frame on the basis of the image number.

9. (Original) A method according to claim 7, further comprising

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. (Original) A method according to claim 7, further comprising

adding the identifier to the Supplemental Enhancement Information (SEI) data structure to be transmitted synchronously with the video sequence.

11. (Original) A method according to claim 1, wherein

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

12. (Original) A method according to claim 1, further comprising

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

13. (Original) A method according to claim 1, further comprising

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

14. (Canceled)

15. (Currently amended) A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of [[the]]other video frames, and the video frames of the second frame format being predicted from at least one ~~of the~~ other video ~~frames~~frame, wherein the video encoder is ~~arranged~~configured to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format; ~~and to~~

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determine a dependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence; and [[to]]

[[form]]encode said dependency into the video sequence ~~information indicating which video frames belong the second sub-sequence~~.

16. (Previously presented) A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

decoding, at a decoder, a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;

decoding, at the decoder, at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determining, at the decoder, dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and

reconstructing, at the decoder, at least part of the video sequence on the basis of the dependencies of the sub-sequences.

17. (Original) A method according to claim 16, wherein video frames are entered into a sliding buffer memory in connection with decoding, further comprising

decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;

configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

18. (Original) A method according to claim 17, further comprising

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

19. (Original) A method according to claim 16, further comprising

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

20. (Original) A method according to claim 16, further comprising

initiating the decoding from a random point in the video sequence;

determining an independently decodable sub-sequence that is next after the random point, and

setting the value of the image number of the first video frame in the sub-sequence at zero.

21. (Original) A method according to claim 16, further comprising

identifying at least partly temporally overlapping sub-sequence frames from the video sequence based on the picture order count information, and

outputting from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.

22. (Currently amended) A video decoder for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the video decoder is ~~arranged~~configured to

decode a first sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the first frame format;

decode at least a second sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.

23. (Currently amended) A computer program product, stored on a computer readable medium and executable in a data processing device, for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; [[and]]

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

a program code for determining a dependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence; and

a program code for ~~determining data~~encoding said dependency into the video sequence ~~to determine which video frames belong to the second sub-sequence~~.

24. (Previously presented) A computer program product, stored on a computer readable medium and executable in a data processing device, for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, wherein the computer program comprises

a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.

25. (Canceled)

26. (Previously presented) A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising

encoding, at an encoder, the video sequence as at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format;

encoding, at the encoder, the video sequence as at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

encoding, at the encoder, into the video sequence information indicating which video frames belong to the second sub-sequence, wherein subsequence information is different from picture type information.

27. (Previously presented) A method according to claim 26, wherein

a removal of the second subsequence from the bitstream does not prevent decoding of the bitstream correctly.

REMARKS/ARGUMENTS

The rejections presented in the Office action dated June 23, 2010 (hereinafter Office action) have been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

Applicant appreciates the indication of allowability for claims 17, 18, and 21 but respectfully declines the opportunity to amend such claims at this time in view of the remarks below.

With respect to the § 112, first paragraph, rejection of claims 2-10, Applicant respectfully traverses. The original specification describes indicating interdependencies in at least paragraphs [0050] and [0059]. It is noted that the subject matter of a claim need not be described literally in the Specification (*i.e.*, using the same terms or *in haec verba*) and an applicant is not limited to the nomenclature used in the application as filed. MPEP §§ 608.01(o) and 2163.01. However, the term "interdependency" has been changed to "dependency" in the claims. Support for these changes may be found in the original specification, for example, at paragraph [0033] and in independent claims 16, 22, and 24; therefore, the changes do not introduce new matter. In view of at least the changes to the objected-to term, the rejection is believed to be overcome. Applicant accordingly requests that the rejection be withdrawn.

Applicant respectfully traverses the § 102(b) rejection based upon U.S. Patent No. 6,363,208 to Nitta *et al.* (hereinafter "Nitta") because Nitta was not patented or described in a printed publication in this or a foreign country more than one year prior to the filing date of the instant application for patent in the United States. The present application was filed in the United States on November 29, 2002. Since Nitta was patented on March 26, 2002, less than a year prior to the filing of the instant application in the United States, § 102(b) is not an appropriate statutory basis for rejection. Applicant accordingly requests that the rejection be withdrawn.

Without acquiescing to characterizations of the asserted art, Applicant's claimed subject matter, or to the applications of the asserted art or combinations thereof to Applicant's claimed subject matter, independent claims 1, 15, and 23 have been amended to

include the limitations of dependent claim 2. Since these limitations were present in original claim 2, the amendments do not introduce new matter. Independent claims 16, 22, 24, and 26 already include subject matter of original claim 2. Since the Office action does not assert that Nitta teaches the limitations of claim 2 (claim 2 is not rejected in view of Nitta), each of the claims is believed to be patentable over Nitta. In addition, the cited portion of Nitta (Col. 1, lines 44-56) merely teaches that the group of pictures includes an intra-frame coding picture (I picture), a P picture, and a B picture without any discussion of determining dependency between frames of a second sub-sequence and at least one video frame of a first sub-sequence and encoding the dependency into the video sequence, as now claimed in each of the independent claims. Without correspondence to each of the claimed limitations, a § 102 rejection based on Nitta would be improper. Applicant accordingly requests that the rejection be withdrawn.

Dependent claims 11-13, 19-20, and 27 depend from independent claims 1, 16, and 26, respectively, and also stand rejected as being anticipated by Nitta. While Applicant does not acquiesce with the particular rejections to these dependent claims, including any assertions concerning inherency, these rejections are moot in view of the remarks made above in connection with the independent claims. These dependent claims include all of the limitations of independent claims 1, 16, and 26 and any intervening claims, and recite additional features which further distinguish these claims from the cited reference. Therefore, the rejection of dependent claims 11-13, 19-20, and 27 is improper and should be withdrawn.

Authorization is given to charge Deposit Account No. 50-3581 (TAPA.004PA) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the undersigned attorney of record invites the Examiner to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

Date: September 15, 2010

By: /Erin Nichols Matkaiti/

Erin Nichols Matkaiti
Reg. No. 57,125

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,934	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on September 17, 2010.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action mailed May 25, 2010, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.111.

Claims 1, 2, 4-6, 8-11, 13, and 14 remain pending in the application. Claims 1, 5, 6, 8-10, 13, and 14 have been amended. No claims have been added, and claims 3, 7, 12, 15, and 16 have been canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 6.

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions of claims in the application.

1. (Currently amended) A method for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame ~~that~~ is earlier than the at least one previous ~~reference~~ image frame in decoding order, the method comprising:

encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence;

encoding identifier values for image frames according to a numbering scheme; and

resetting the identifier value for the indicated first image frame of an independent sequence.

2. (Original) A method according to claim 1, further comprising:

encoding the indication into the video sequence as a separate flag included in the header of a slice.

3. (Canceled)

4. (Original) A method according to claim 1, further comprising:

encoding into the video sequence an identifier value for the independent sequence.

5. (Currently amended) A video encoder for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference frame ~~that~~ is earlier than the at least one previous ~~reference~~ image frame in decoding order, the video encoder being ~~arranged~~configured to:

encode into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence;

encode identifier values for image frames according to a numbering scheme; and

reset the identifier value for the indicated first image frame of an independent sequence.

6. (Currently amended) A video encoder according to claim 5, wherein the video encoder is ~~arranged~~configured to

encode the indication into the video sequence as a separate flag included in the header of a slice.

7. (Canceled)

8. (Currently amended) A video encoder according to claim 5, wherein the video encoder is ~~arranged~~configured to

encode into the video sequence an identifier value for the independent sequence.

9. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame [[that]]and wherein the at least one reference frame is earlier than the at least one previous ~~reference~~ image frame in decoding order, the computer program product comprising:

a computer program code for encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence;

a computer code for encoding identifier values for image frames according to a numbering scheme; and

a computer code for resetting the identifier value for the indicated first image frame of an independent sequence.

10. (Currently amended) A method for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame [[that]]and wherein the at least one reference frame is earlier than the at least one previous ~~reference~~ image frame in decoding order, the method comprising:

decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame; [[and]]

starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;

decoding identifier values for image frames according to a numbering scheme; and

resetting the identifier value for the indicated first image frame of an independent sequence.

11. (Original) A method according to claim 10, wherein the indication is a separate flag included in the header of a slice.

12. (Canceled)

13. (Currently amended) A video decoder for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame ~~that~~ is earlier than the at least one previous ~~reference~~ image frame in decoding order, the video decoder being ~~arranged~~configured to:

decode from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame; [[and]]

start the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame; and

decode identifier values for image frames according to a numbering scheme; and

reset the identifier value for the indicated first image frame of an independent sequence.

14. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame ~~that~~ is earlier than the at least one previous ~~reference~~ image frame in decoding order, the computer program product comprising:

a computer program code for decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame; [[and]]

a computer program code for starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;

a computer program code for decoding identifier values for image frames according to a numbering scheme; and

a computer program code for resetting the identifier value for the indicated first image frame of an independent sequence.

15-16. (Canceled)

REMARKS/ARGUMENTS

The rejections presented in the Office action dated May 25, 2010 (hereinafter Office Action), have been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

Claims 3, 7, 12, 15, and 16 have been canceled, without prejudice, rendering the respective rejections of these claims moot. Applicant accordingly requests that the rejections of each of these claims be withdrawn.

With respect to the objection to claims 1, 2, 4, 5, 6, 8, 10, and 11, the term "the previous reference image frame" in independent claims 1, 5, and 10 has been amended to read "the previous image frame" in order to provide antecedent basis for the term. Applicant accordingly requests that the objection be removed.

With respect to the objection to Fig. 1, Applicant respectfully traverses. In contrast to the assertion that the figure includes "blank boxes", each of the boxes of Fig. 1 is labeled with a reference numeral in accordance with MPEP § 608.02. Moreover, no requirement has been cited for the request to label the boxes with text. Since Fig. 1 complies with the drawing requirements, and in an effort to maintain consistency with the parent application, Applicant requests that the objection be removed.

With respect to the § 101 rejection of claims 5-8 and 13, Applicant respectfully traverses. The Office action incorrectly asserts that the claimed encoder and decoder are "pure software". While the specification refers to embodiments involving computer readable media storing program code (*e.g.*, claims 9 and 14), the claimed encoder and decoder are devices constituting statutory subject matter. The terms "encoder" and "decoder" are well-known in the art to refer to a device or circuit. Notably, the specification illustrates structural components of the encoder (*e.g.*, compressor 108 and multiplexer 110) and decoder (*e.g.*, demultiplexer 116 and decompressor 118). Paragraph [0095] teaches examples of a decoder including a low bit rate video decoder as described in the ITU-T recommendation H.263 or H.26L. Moreover, paragraph [0096] describes positioning of the encoder and decoder that would not be possible if they were "pure

software". Contrary to the assertion, the claimed encoder and decoder are devices constituting statutory subject matter. Applicant accordingly requests that the rejection be withdrawn.

With respect to the § 101 rejection of claims 9 and 14, the claims have been amended to indicate that the claimed computer readable media are non-transitory in accordance with the Examiner's suggestion. Applicant notes that non-transitory computer readable media comprise all computer readable media, with the sole exception being a transitory, propagating signal. Since an interpretation of the claims is now limited to statutory subject matter, Applicant submits that the rejection has been overcome and accordingly requests that the rejection be withdrawn.

Applicant respectfully traverses the § 112, first paragraph, rejection of claims 5, 6, 8, and 13 because the claims are not "single means" claims. First, none of these claims uses a "means recitation" or "means plus function" language. While a claim dependent upon a recited property could result in a fact situation similar to that discussed in the cited *Hyatt* decision when the claim covers every conceivable structure for achieving the stated property, such is not the case for claims 5, 6, 8, and 13. In contrast the claims are limited to the structures of an encoder (claims 5, 6, and 8) and of a decoder (claim 13), which are well-accepted and understood terms in the art, for achieving the stated properties of the respective claims. Therefore, the claims are not unduly broad and are enabled by the specification. Applicant accordingly requests that the rejection be withdrawn.

Applicant respectfully traverses the § 102(e) rejection of claim 1 in view of U.S. Publication No. 2003/0063806 by Kim *et al*. (hereinafter "Kim") because the Office action acknowledges that Kim fails to teach or suggest each of the claimed limitations. In the § 103(a) rejection of claim 1 at page fifteen, the Office action states that "Kim fails to explicitly disclose an indication of the first I frame". Since Kim admittedly fails to correspond to each of the limitations of at least claim 1, the § 102(e) rejection is improper. Applicant accordingly requests that the rejection be withdrawn.

In an effort to facilitate prosecution and without acquiescing to characterizations of the asserted art, Applicant's claimed subject matter, or to the applications of the asserted art

or combinations thereof to Applicant's claimed subject matter, the independent claims have been amended to include the respective limitations of dependent claims 3, 7, and 12; therefore, these changes do not introduce new matter. Since the Office action acknowledges that Kim and U.S. Patent No. 6,094,456 to Ueda fail to teach or suggest each of the limitations of dependent claims 3, 7, and 12, each of the rejections based on these references is overcome. Each of the claims is further believed to be patentable over the asserted references for the reasons set forth below.

In rejecting dependent claims 3, 7, and 12, the Office action relies upon the teachings of U.S. Patent No. 4,390,966 to Kawashima *et al.* (hereinafter "Kawashima") as teaching the claimed resetting of the identifier value for the indicated first image frame of an independent sequence; however, this reliance is misplaced. Kawashima merely teaches that a sequence controller system is capable of setting and resetting flags according to predetermined user instructions. Kawashima's flags fail to at least correspond to the claimed identifier value since Kawashima is silent with respect to the flags being associated with an image frame of a coded video sequence. Also, Kawashima's flags are toggle flags for use in a multi CPU environment where each controller can run sequentially stored code in a parallel environment. There is no correspondence to the claimed resetting of an encoded identifier value for a first image frame of a video sequence. Without correspondence to each of the claimed limitations, the § 103(a) rejections of the subject matter of dependent claims 3, 7, and 12 is improper. Applicant accordingly requests that the rejections be withdrawn.

Further, the asserted modification of the combination of Kim and Ueda with the teachings of Kawashima is improper since the asserted modification is illogical and the teachings of Kawashima are not analogous to the field of Applicant's endeavor. First, the Office action asserts that it would be desirable to reset the identifier value since the video sequence is independent because it would simplify the identifier value and reset the identifier value. No explanation or evidence has been provided to indicate that resetting Kawashima's flags would simplify a value. Moreover, it is illogical to assert that it is

desirable to reset an identifier value because it would reset the identifier value. Thus, the proposed modification to the combination of Kim and Ueda is illogical and unsupported.

Second, the teachings of Kawashima are nonanalogous art as they are directed to very different technology than the field of video coding. In contrast, Kawashima is directed to the field of multi-core processing. Since Kawashima is in a field different than Applicant's endeavor, to be considered reasonably pertinent, Kawashima would need to address a problem addressed by the application at issue (MPEP § 2141.01(a)). However, the Office action fails to assert or identify any such relationship. Kawashima, directed to very different technology and failing to address video coding, would not have logically commended itself to an inventor of the instant application. Thus, Kawashima is nonanalogous art and fails to support the § 103(a) rejections. Applicant respectfully maintains the request that the § 103(a) rejections based upon Kawashima be withdrawn.

It should also be noted that Applicant does not acquiesce to the Examiner's statements or conclusions concerning what would have been obvious to one of ordinary skill in the art, obvious design choices, common knowledge at the time of Applicant's invention, officially noticed facts, and the like. Applicant reserves the right to address in detail the Examiner's characterizations, conclusions, and rejections in future prosecution.

With respect to the provisional rejections of claims 1-4 under the judicially-created doctrine of obviousness-type double patenting over co-pending and commonly-owned patent application no. 10/306,942 (hereinafter "the '942 application"), Applicant will consider filing a Terminal Disclaimer once the instant application is in condition for allowance or the '942 application issues as a patent. However, Applicant respectfully asserts that, in view of the amendments to the claims and arguments made above, the Examiner is compelled to withdraw the rejections of the claims. Once withdrawn, the only rejection remaining in the instant application would be the provisional obviousness-type double patenting rejection. In view of MPEP § 804 I(B), Applicant respectfully requests that the provisional obviousness-type double patenting rejection be withdrawn and that the instant application be permitted to issue as a patent.

Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C2) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the undersigned attorney of record invites the Examiner to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

Date: September 17, 2010

By: /Erin Nichols Matkaiti/

Erin Nichols Matkaiti
Reg. No. 57,125

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. § 1.97(c)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant respectfully submits the items of information on the enclosed Form 1449 for the attention of the Examiner in the above-identified application.

This statement should be considered because it is submitted after the mailing date of the first Office action on the merits but before the mailing date of a final Office action under 37 C.F.R. § 1.113 or a Notice of Allowance under 37 C.F.R. § 1.311. Please charge Deposit Account 50-3581 (TPO.004.C1) the amount of $180.00 under 37 C.F.R. § 1.17(p) for consideration of the items.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. § 1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti
Reg. No. 57,125

Electronic Patent Application Fee Transmittal

Application Number:	11338996
Filing Date:	25-Jan-2006
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Filer:	Erin M. Nichols
Attorney Docket Number:	TPO.004.C1

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
		Total in USD ($)		**180**

Electronic Acknowledgement Receipt

EFS ID:	8531052
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	30-SEP-2010
Filing Date:	25-JAN-2006
Time Stamp:	15:11:54
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$180
RAM confirmation Number	4430
Deposit Account	503581
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Filed (SB/08)	1449-11-TPO004C1.pdf	51553 3ec91342c58482a63da3b9b20c7843f934fc2151	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
2	NPL Documents	OAR-TAPA004PA-OA23062010.pdf	97281 8e13d6c942b64fe9bc59ac82106b7e9c0b09ce2a	no	12
Warnings:					
Information:					
3	NPL Documents	OAR-TPO004C2-OA25052010.pdf	95492 e2dea1ff0412ff322887c375b928c5a16e435a48	no	10
Warnings:					
Information:					
4	Transmittal Letter	IDS-11-TPO004C1.pdf	55421 e891388981d58287aa2f1a6f89d230208867bd33	no	2
Warnings:					
Information:					
5	Fee Worksheet (PTO-875)	fee-info.pdf	30216 78ab6c0bc3ccbb181ea97a0683a6a660f45f3069	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	329963		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	JP11275576	10/08/1999	JP			Abstract only	
	JP4348690	12/03/1992	JP			Abstract only	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	08/17/2010	Copy of translated Office Action dated 08/17/2010 from parallel Japanese Application No. 2006-120297, 4 pages.
	08/24/2010	Copy of translated Office Action dated 08/24/2010 from parallel Japanese Application No. 2006-120296, 5 pages.

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

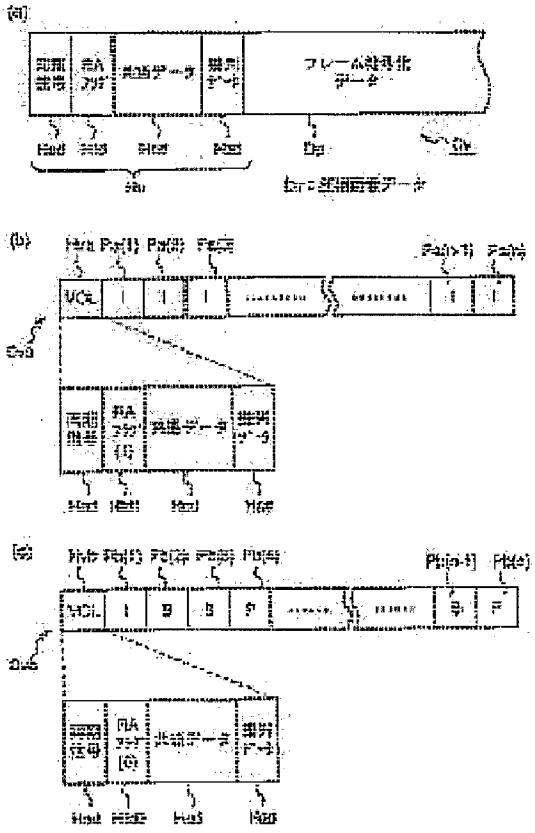
IMAGE TRANSMISSION METHOD, IMAGE PROCESSING METHOD, IMAGE PROCESSOR AND DATA STORAGE MEDIUM

Publication number: JP11275576 (A)
Publication date: 1999-10-08
Inventor(s): BUN CHUN SEN +
Applicant(s): MATSUSHITA ELECTRIC IND CO LTD +
Classification:
- international: ***H04N7/24; H04N7/26; H04N7/32;*** **H04N7/24; H04N7/26; H04N7/32;** (IPC1-7): H04N7/24
- European:
Application number: JP19990014735 19990122
Priority number(s): JP19990014735 19990122; JP19980011068 19980123

Also published as:
JP3380763 (B2)

Abstract of **JP 11275576 (A)**

PROBLEM TO BE SOLVED: To detect aptitude of random independent reproduction processing in a short time with respect to compressed image data through the analysis of an identification flag in the case of analysis processing of a header imparted to the compressed image data · corresponding to a moving image. SOLUTION: In the image transmission method, compressed image data Dv obtained by applying compression coding processing to digital image data corresponding to a moving image are transmitted, and in this case, an identification flag Hfd denoting whether or not the compressed image data Dv are proper to random independent reproduction onto an optional screen is sent next to a synchronizing signal Hsd placed at a head of a header Hv.



Data supplied from the ***espacenet*** database — Worldwide

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平11－275576

(43)公開日　平成11年(1999)10月８日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｈ０４Ｎ　7/24		Ｈ０４Ｎ　7/13	Ｚ

審査請求　未請求　請求項の数22　ＯＬ　（全 31 頁）

(21)出願番号　特願平11－14735

(22)出願日　平成11年(1999)１月22日

(31)優先権主張番号　特願平10－11068

(32)優先日　平10(1998)１月23日

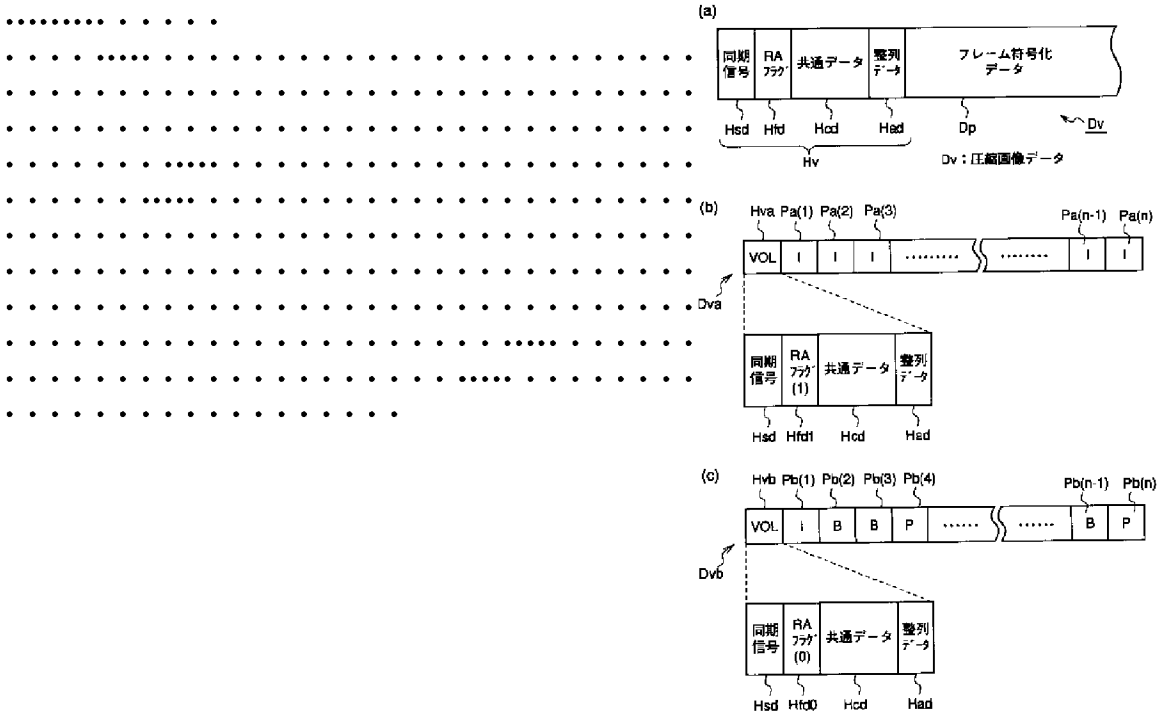
(33)優先権主張国　日本（ＪＰ）

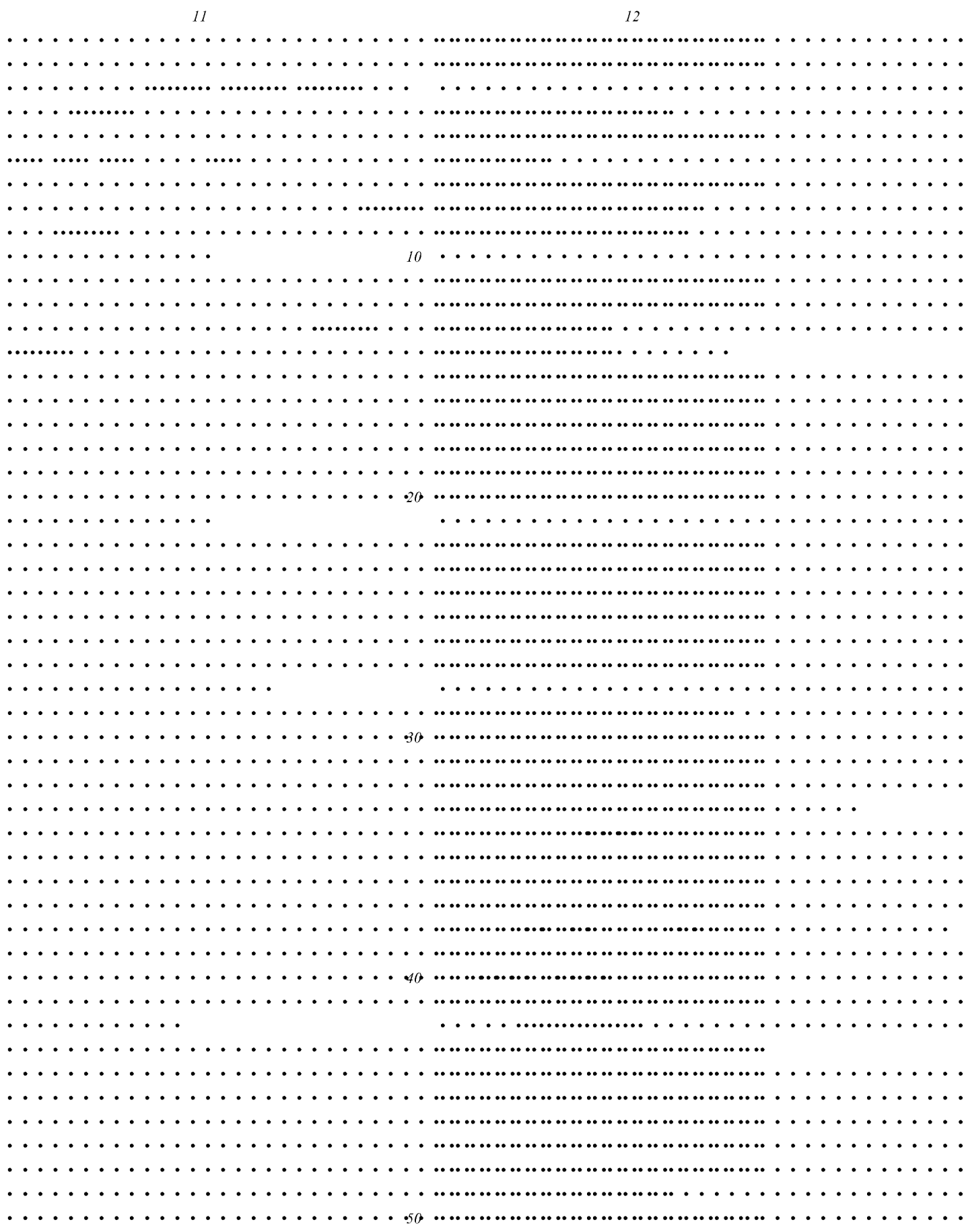
(71)出願人　000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地

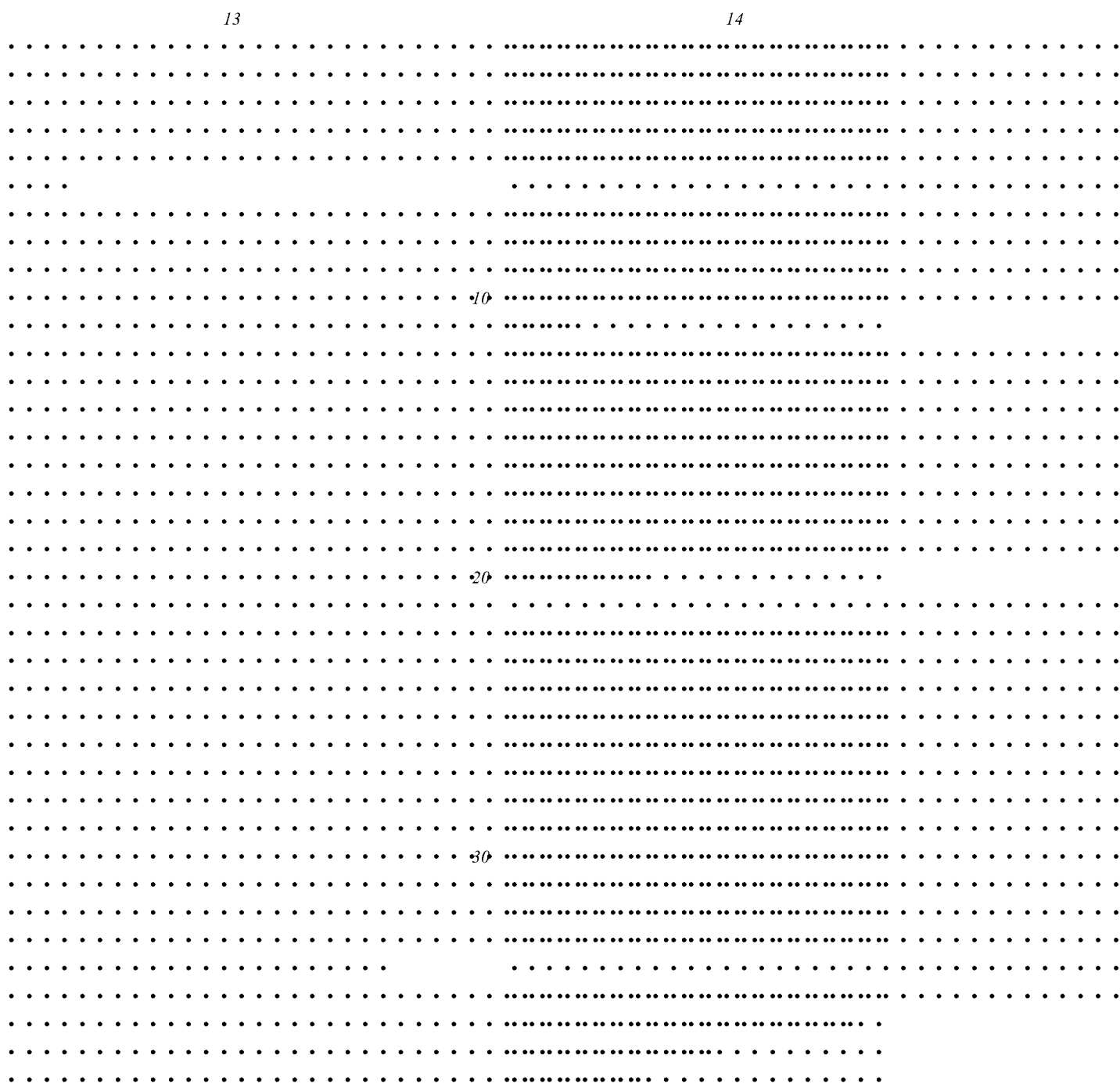
(72)発明者　ブン　チュン　セン
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(74)代理人　弁理士　早瀬　憲一

(54)【発明の名称】　画像伝送方法，画像処理方法，画像処理装置，及びデータ記憶媒体



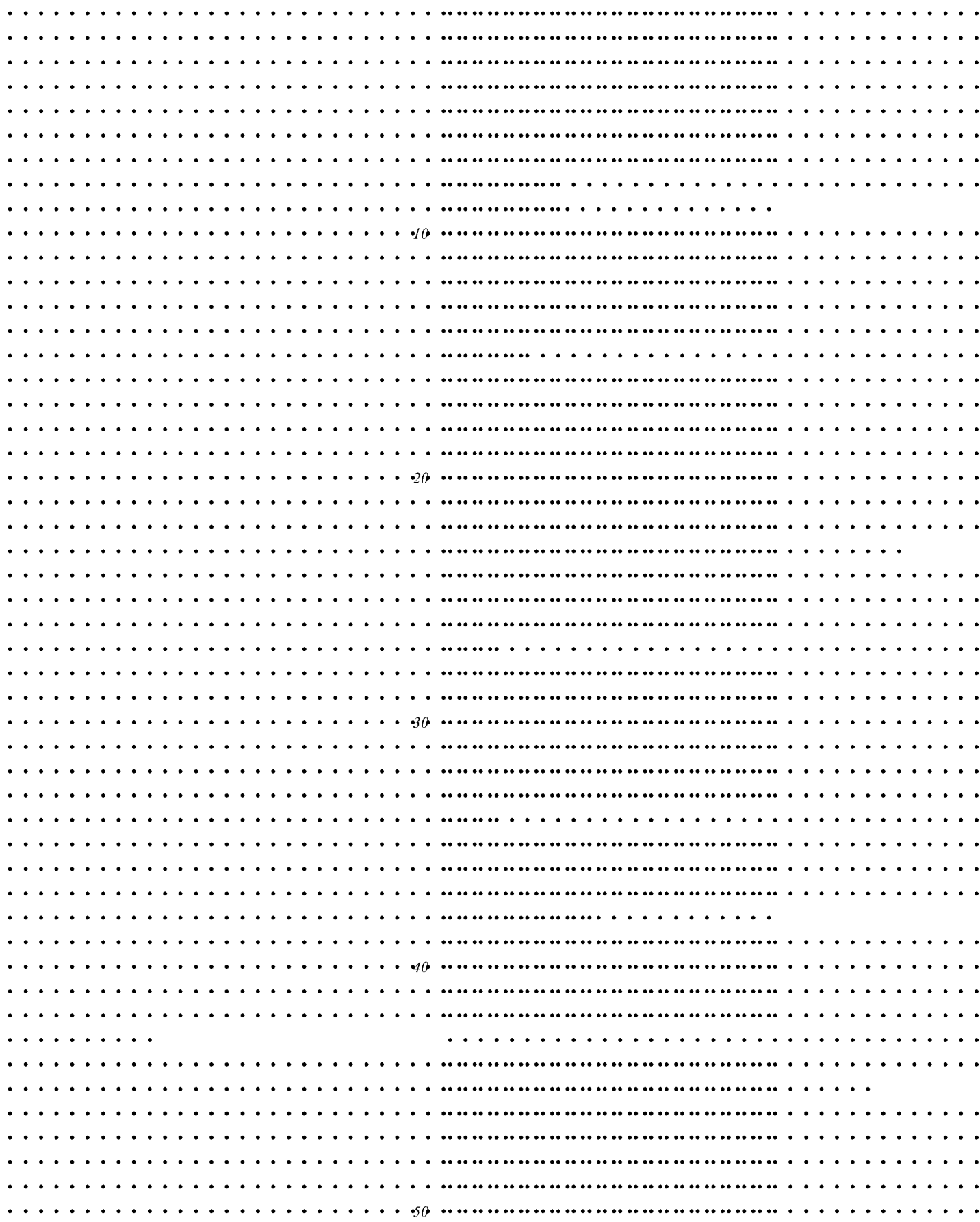

11
12
10
20
30
40
50


13
14
10
20
30

		No.of bits	Mnemonic
901	Video Object Layer(){		
902	video_object_layer_start_code /*4least significant bits specify video_object_layer_id value*/	32	bslbf
903	is_object_layer_identifier ~903a	1	uimsbf
	if(is_object_layer_identifier){		
	visual_object_layer_verid ~903b	4	uimsbf
	visual_object_layer_priority ~903c	3	uimsbf
	}		
904	vol_control_parameters	1	bslbf
	if(vol_control_parameters)		
	aspect_ratio_info	4	uimsbf
	vop_rate_code	4	uimsbf
	bit_rate	30	uimsbf
	vbv_buffer_size	18	uimsbf
	chroma_format	2	uimsbf
	low_delay	1	uimsbf
	}		
905	video_object_layer_shape	2	uimsbf
	vop_time_increment_resolution	15	uimsbf
	fixed_vop_rate	1	bslbf
906	if(video_object_layer_shape!="binary only"){		
907	if(video_object_layer_shape=="rectangular"){		
	marker_bit	1	bslbf
	video_object_layer_width	13	uimsbf
	marker_bit	1	bslbf
	video_object_layer_height	13	uimsbf
	}		
	obmc_disable	1	bslbf

908	sprite_enable	1	bslbf
	if(sprite_enable){		
	sprite_width	13	uimsbf
	marker_bit	1	bslbf
	sprite_height	13	uimsbf
	marker_bit	1	bslbf
	sprite_left_coodinate	13	simsbf
	marker_bit	1	bslbf
	sprite_top_coodinate	13	simsbf
	marker_bit	1	bslbf
	no_of_sprite_warping_points	6	uimsbf
	sprite_warping_accuracy	2	uimsbf
	sprite_brightness_change	1	bslbf
	if(video_object_layer_shape=="rectangular"){		
	init_sprite_width	13	uimsbf
	marker_bit	1	bslbf
	init_sprite_height	13	uimsbf
	marker_bit	1	bslbf
	init_sprite_left_coodinate	13	simsbf
	marker_bit	1	bslbf
	init_sprite_top_coodinate	13	simsbf
	}		
	}		
909	not_8_bit	1	bslbf
	if(not_8_bit){		
	quant_precision	4	uimsbf
	bits_per_pixel	4	uimsbf
	}		
910	quant_type	1	bslbf
	if(quant_type){		
	load_intra_quant_mat	1	bslbf
	if(load_intra_quant_mat)		
	intra_quant_mat	8*[2-64]	uimsbf
	load_nonintra_quant_mat	1	bslbf
	if(load_nonintra_quant_mat)		
	nonintra_quant_mat	8*[2-64]	uimsbf
	}		
911	complexity_estimation_disable	1	bslbf

912	error_resilient_disable	1	bslbf
	if(!error_resilient_disable){		
	data_partitioned	1	bslbf
	reversible_vlc	1	bslbf
	}		
913	scalability	1	bslbf
	if(scalability){		
	ref_layer_id	4	uimsbf
	ref_layer_sampling_direc	1	bslbf
	hor_sampling_factor_n	5	uimsbf
	hor_sampling_factor_m	5	uimsbf
	vert_sampling_factor_n	5	uimsbf
	vert_sampling_factor_m	5	uimsbf
	enhancement_type	1	bslbf
	}		
	}		
914	random_accessible_vol	1	bslbf
915	next_start_code()		
916	if(sprite_enable)		
	decode_init_sprite()		
917	do{		
	if(next_bits()=group_of_vop_start_code)		
	Group_of_Video object Plane()		
	Video Object Plane()		
	}while((next_bits()=group_of_vop_start_code)|| (next_bits()=vop_start_code)		
	}		

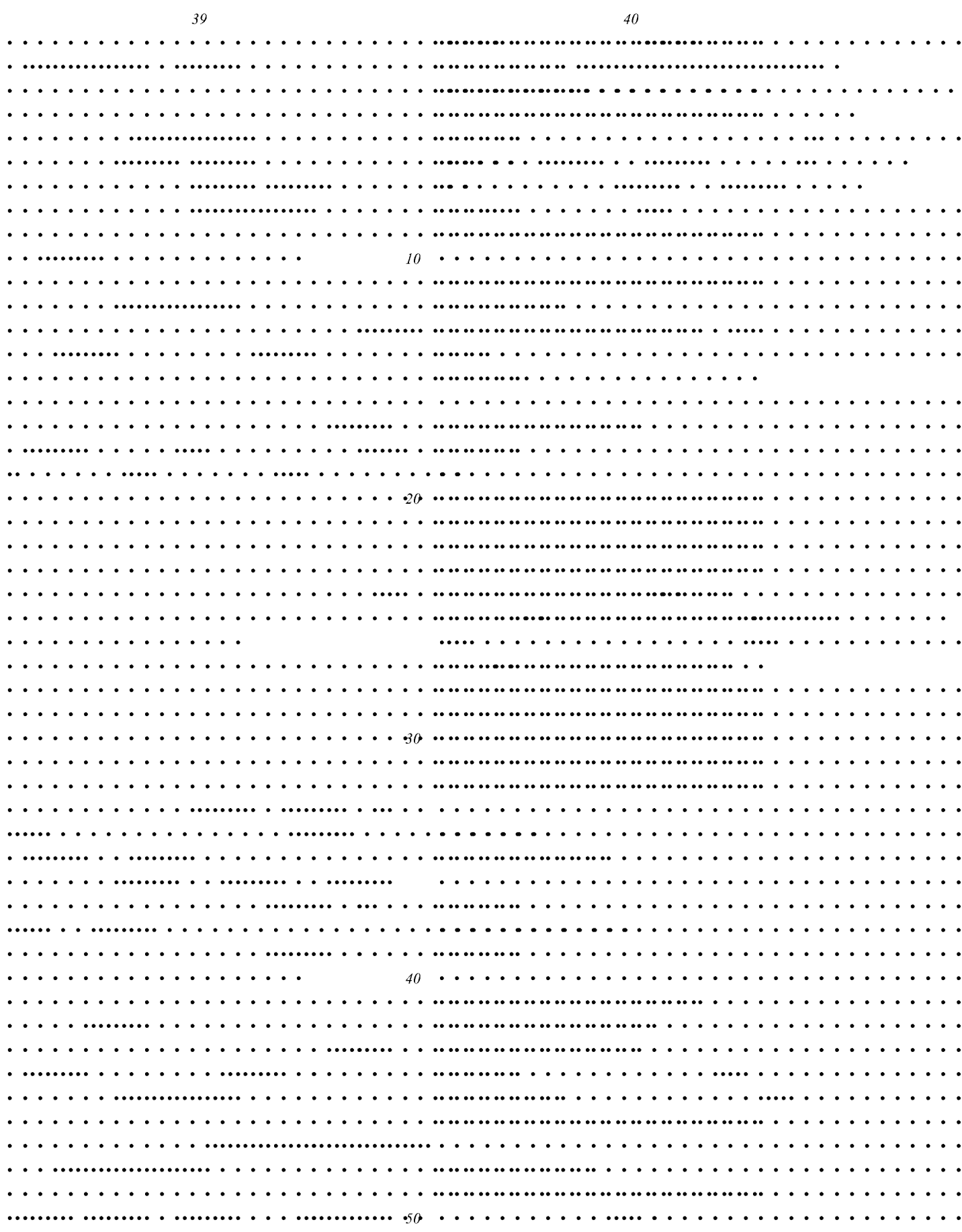


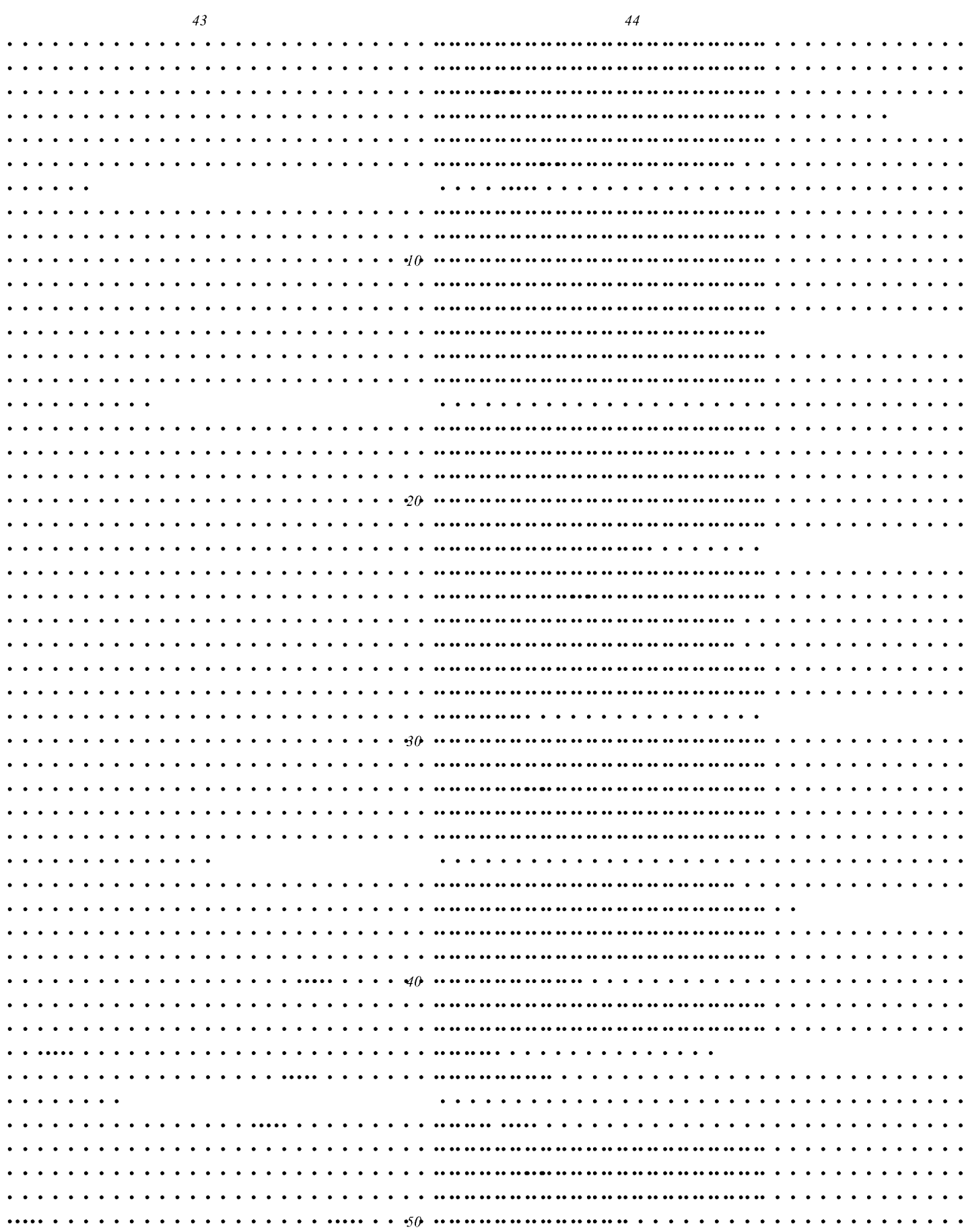
	Video Object Layer(){	No.of bits	Mnemonic
802	video_object_layer_start_code /*4least significant bits specify video_object_layer_id value*/	32	bslbf
814	random_accessible_vol	1	bslbf
803	is_object_layer_identifier	1	uimsbf
	if(is_object_layer_identifier){		
	visual_object_layer_verid	4	uimsbf
	visual_object_layer_priority	3	uimsbf
	}		
804	vol_control_parameters	1	bslbf
	if(vol_control_parameters)		
	aspect_ratio_info	4	uimsbf
	vop_rate_code	4	uimsbf
	bit_rate	30	uimsbf
	vbv_buffer_size	18	uimsbf
	chroma_format	2	uimsbf
	low_delay	1	uimsbf
	}		
805	video_object_layer_shape	2	uimsbf
	vop_time_increment_resolution	15	uimsbf
	fixed_vop_rate	1	bslbf
806	if(video_object_layer_shape!="binary only"){		
807	if(video_object_layer_shape=="rectangular"){		
	marker_bit	1	bslbf
	video_object_layer_width	13	uimsbf
	marker_bit	1	bslbf
	video_object_layer_height	13	uimsbf
	}		
	obmc_disable	1	bslbf

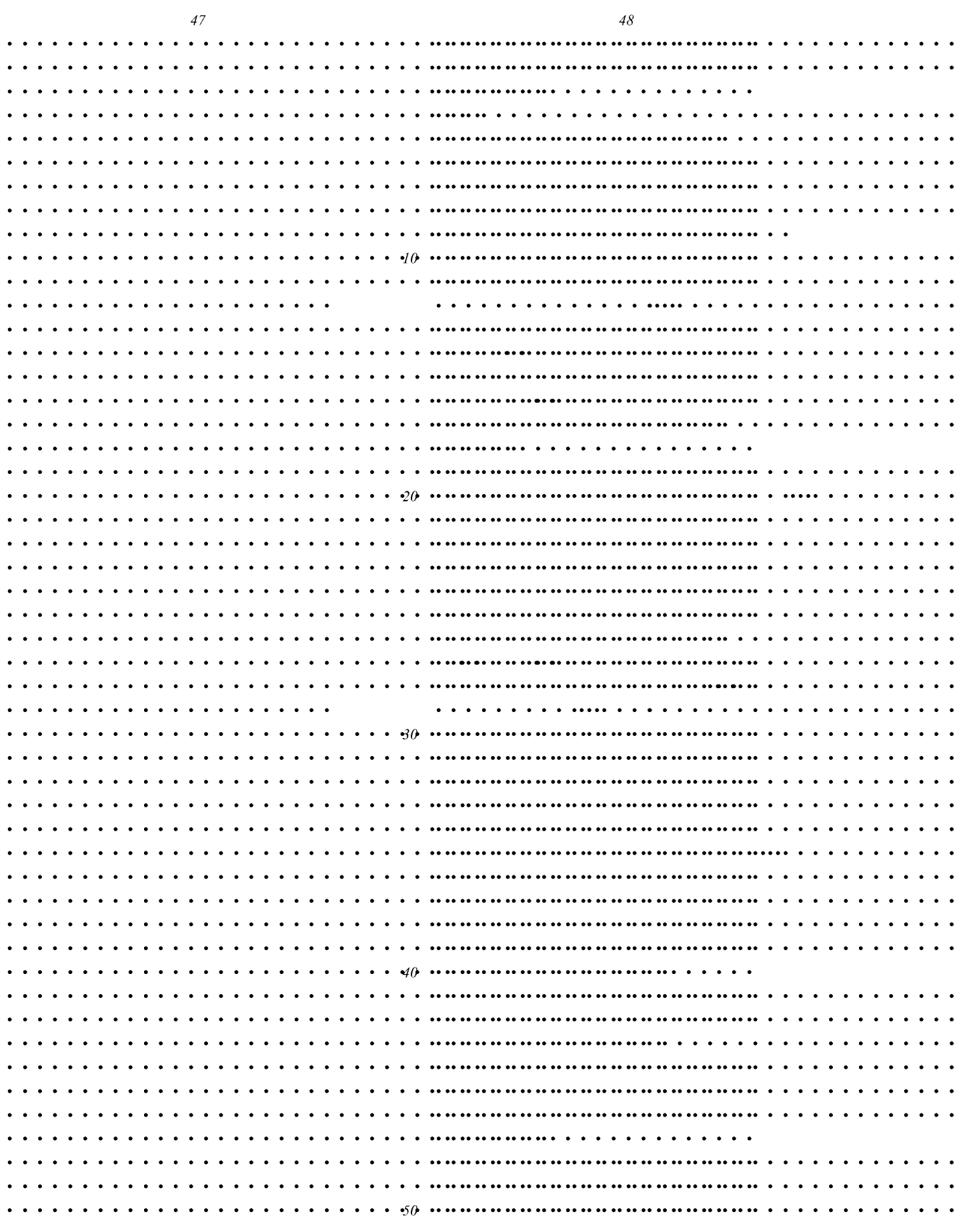
31 32

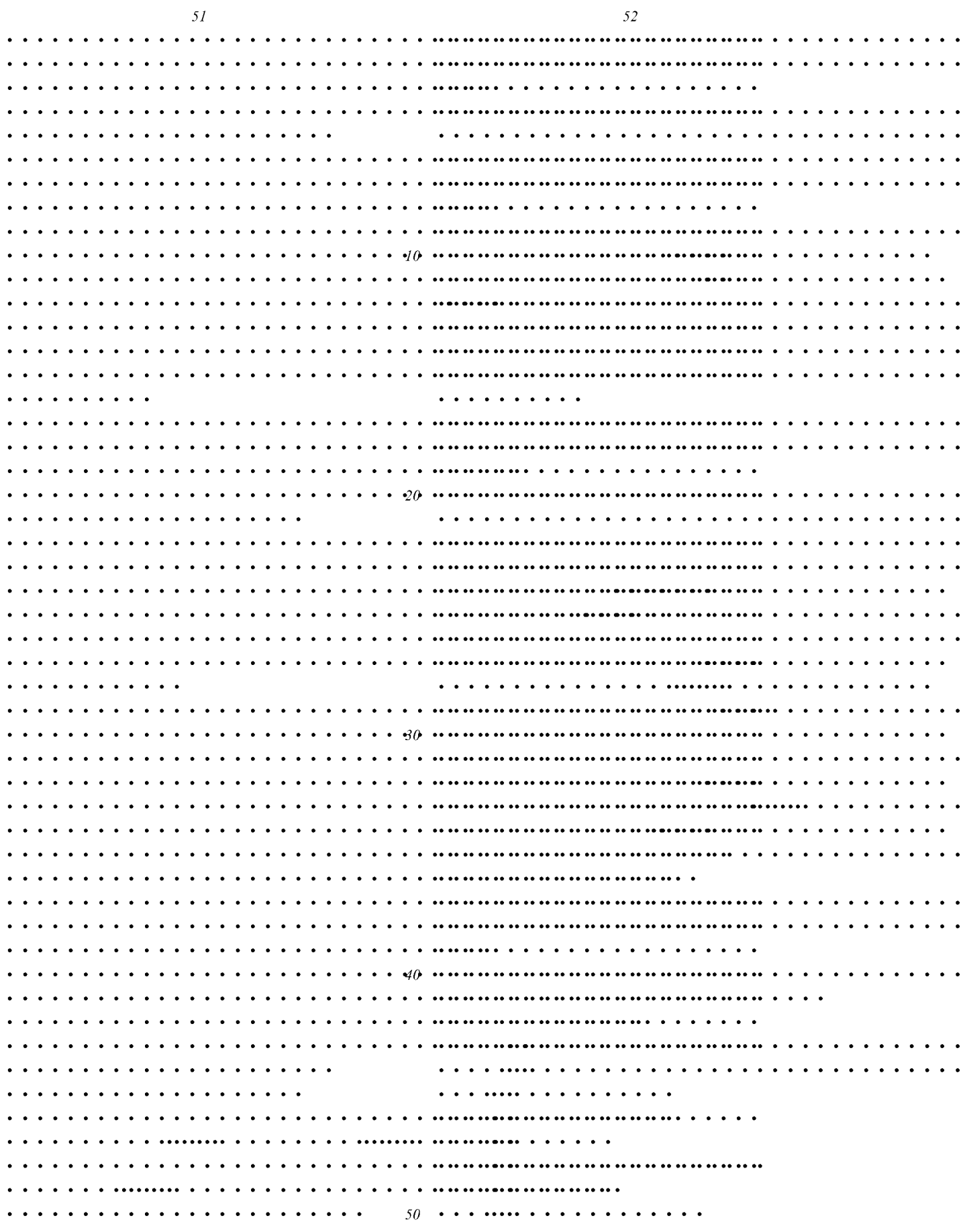
808	sprite_enable	1	bslbf
	if(sprite_enable){		
	sprite_width	13	uimsbf
	marker_bit	1	bslbf
	sprite_height	13	uimsbf
	marker_bit	1	bslbf
	sprite_left_coodinate	13	simsbf
	marker_bit	1	bslbf
	sprite_top_coodinate	13	simsbf
	marker_bit	1	bslbf
	no_of_sprite_warping_points	6	uimsbf
	sprite_warping_accuracy	2	uimsbf
	sprite_brightness_change	1	bslbf
	if(video_object_layer_shape=="rectangular"){		
	init_sprite_width	13	uimsbf
	marker_bit	1	bslbf
	init_sprite_height	13	uimsbf
	marker_bit	1	bslbf
	init_sprite_left_coodinate	13	simsbf
	marker_bit	1	bslbf
	init_sprite_top_coodinate	13	simsbf
	}		
	}		
809	not_8_bit	1	bslbf
	if(not_8_bit){		
	quant_precision	4	uimsbf
	bits_per_pixel	4	uimsbf
	}		
810	quant_type	1	bslbf
	if(quant_type){		
	load_intra_quant_mat	1	bslbf
	if(load_intra_quant_mat)		
	intra_quant_mat	8*[2-64]	uimsbf
	load_nonintra_quant_mat	1	bslbf
	if(load_nonintra_quant_mat)		
	nonintra_quant_mat	8*[2-64]	uimsbf
	}		
811	complexity_estimation_disable	1	bslbf

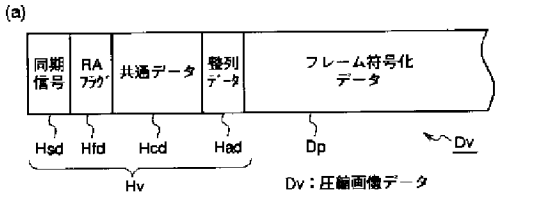
812	error_resilient_disable	1	bslbf
	if(!error_resilient_disable){		
	data_partitioned	1	bslbf
	reversible_vlc	1	bslbf
	}		
813	scalability	1	bslbf
	if(scalability){		
	ref_layer_id	4	uimsbf
	ref_layer_sampling_direc	1	bslbf
	hor_sampling_factor_n	5	uimsbf
	hor_sampling_factor_m	5	uimsbf
	vert_sampling_factor_n	5	uimsbf
	vert_sampling_factor_m	5	uimsbf
	enhancement_type	1	bslbf
	}		
	}		
815	next_start_code()		
816	if(sprite_enable)		
	decode_init_sprite()		
817	do{		
	if(next_bits()=group_of_vop_start_code)		
	Group_of_Video object Plane()		
	Video Object Plane()		
	}while((next_bits()=group_of_vop_start_code)|| (next_bits()=vop_start_code)		
	}		

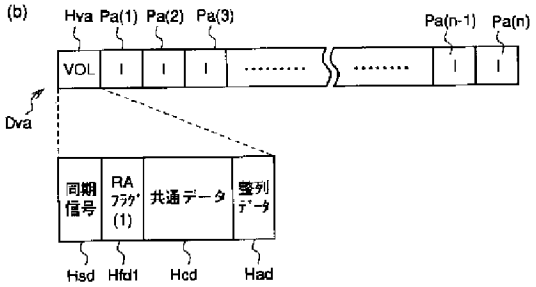

39
40
10
20
30
40
50

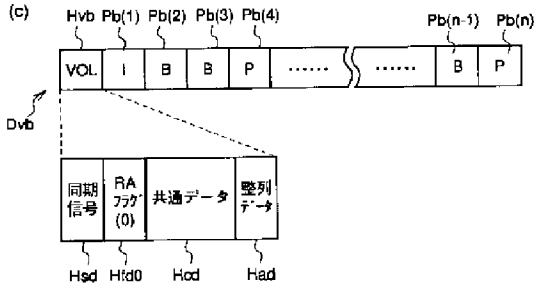


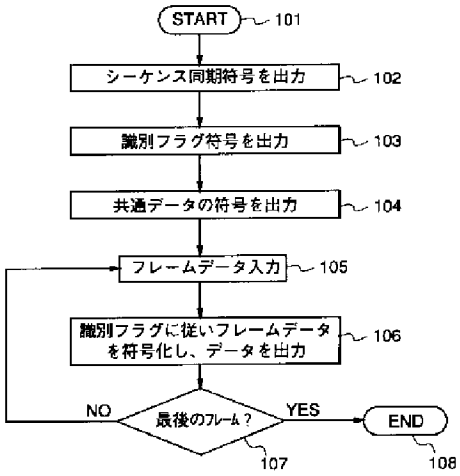


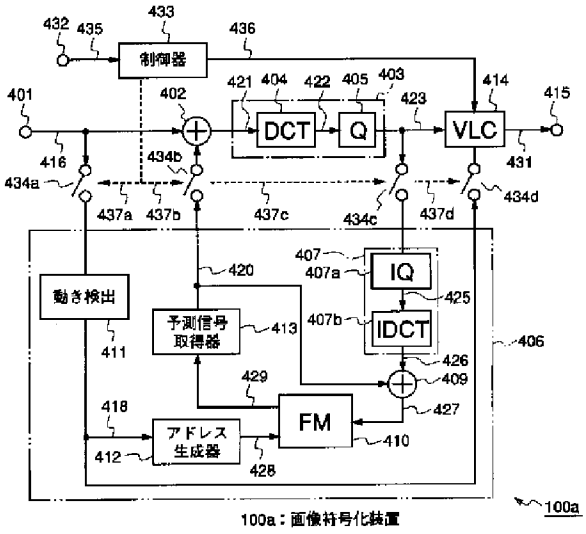

51
52
10
20
30
40
50

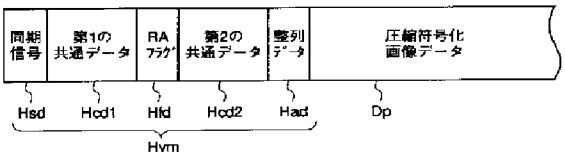

(a)
同期信号
RA フラグ
共通データ
整列データ
フレーム符号化データ
Hsd
Hfd
Hcd
Had
Dp
Dv
Hv
Dv：圧縮画像データ

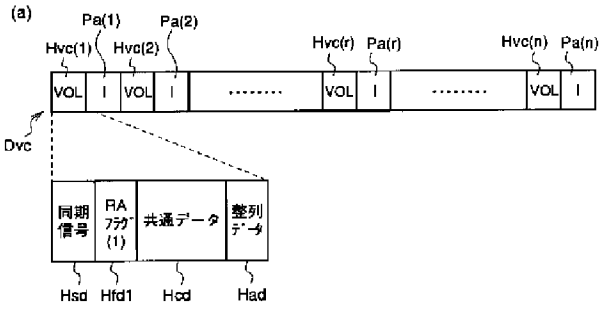

(b)
Hva Pa(1) Pa(2) Pa(3)
Pa(n-1) Pa(n)
VOL
I
Dva
同期信号
RA フラグ (1)
共通データ
整列データ
Hsd Hfd1 Hcd Had

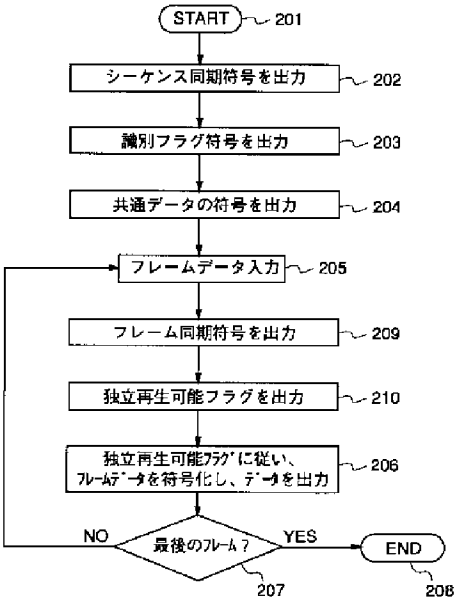

(c)
Hvb Pb(1) Pb(2) Pb(3) Pb(4)
Pb(n-1) Pb(n)
VOL
I
B
P
Dvb
同期信号
RA フラグ (0)
共通データ
整列データ
Hsd Hfd0 Hcd Had

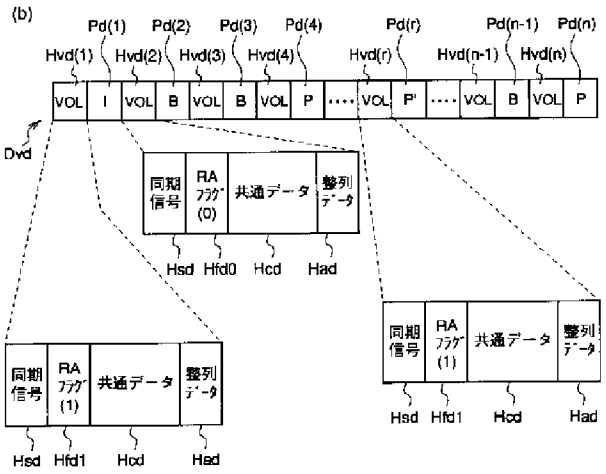

START
101
シーケンス同期符号を出力
102
識別フラグ符号を出力
103
共通データの符号を出力
104
フレームデータ入力
105
識別フラグに従いフレームデータを符号化し、データを出力
106
最後のフレーム？
NO
YES
107
END
108

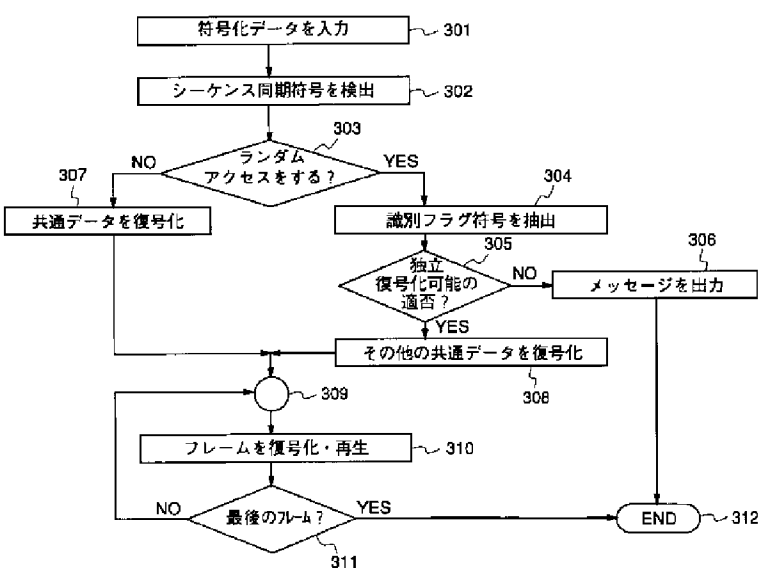

制御器
DCT
Q
VLC
IQ
IDCT
FM
動き検出
予測信号取得器
アドレス生成器
100a
100a：画像符号化装置

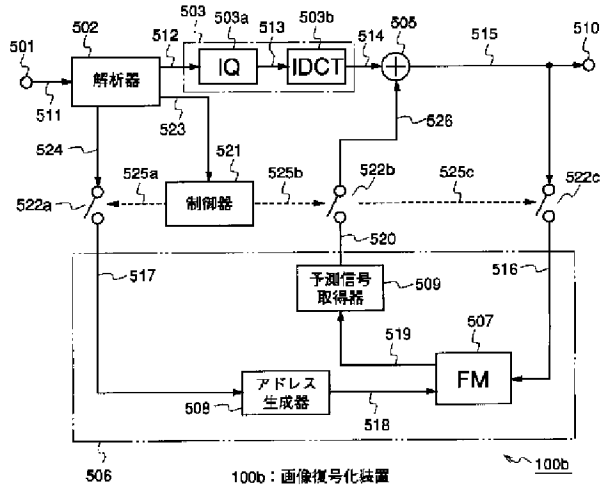

同期信号
第1の共通データ
RAフラグ
第2の共通データ
整列データ
圧縮符号化画像データ
Hsd
Hcd1
Hfd
Hcd2
Had
Dp
Hvm

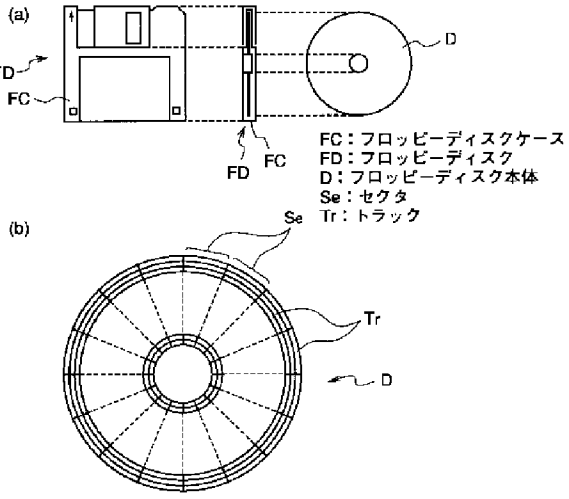

(a)
Pa(1)
Pa(2)
Hvc(1)
Hvc(2)
Hvc(r)
Pa(r)
Hvc(n)
Pa(n)
VOL
I
Dvc
同期信号
RAフラグ (1)
共通データ
整列データ
Hsd
Hfd1
Hcd
Had

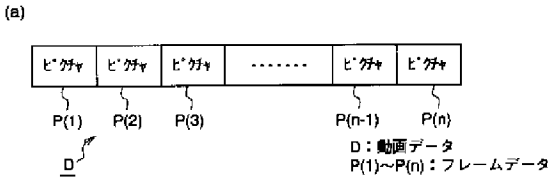

START
201
シーケンス同期符号を出力
202
識別フラグ符号を出力
203
共通データの符号を出力
204
フレームデータ入力
205
フレーム同期符号を出力
209
独立再生可能フラグを出力
210
独立再生可能フラグに従い、フレームデータを符号化し、データを出力
206
NO
最後のフレーム？
YES
END
207
208

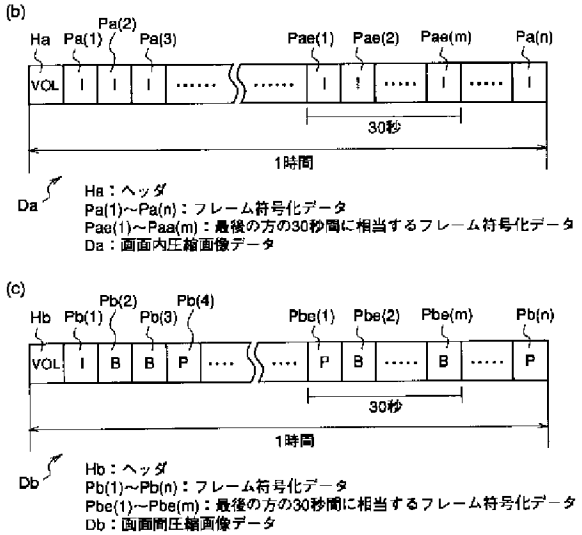

(b)
Pd(1)
Pd(2)
Pd(3)
Pd(4)
Pd(r)
Pd(n-1)
Pd(n)
Hvd(1)
Hvd(2)
Hvd(3)
Hvd(4)
Hvd(r)
Hvd(n-1)
Hvd(n)
VOL
I
B
P
P'
Dvd
同期信号
RAフラグ (0)
共通データ
整列データ
Hsd
Hfd0
Hcd
Had
同期信号
RAフラグ (1)
共通データ
整列データ
Hsd
Hfd1
Hcd
Had

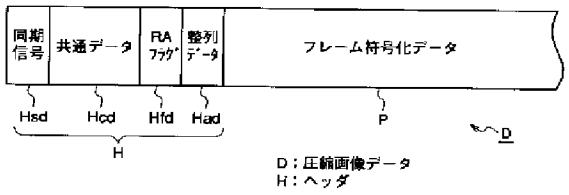

符号化データを入力
301
シーケンス同期符号を検出
302
303
ランダムアクセスをする？
NO
YES
307
共通データを復号化
304
識別フラグ符号を抽出
305
独立復号化可能の適否？
NO
306
メッセージを出力
YES
その他の共通データを復号化
308
309
フレームを復号化・再生
310
NO
最後のフレーム？
YES
END
312
311

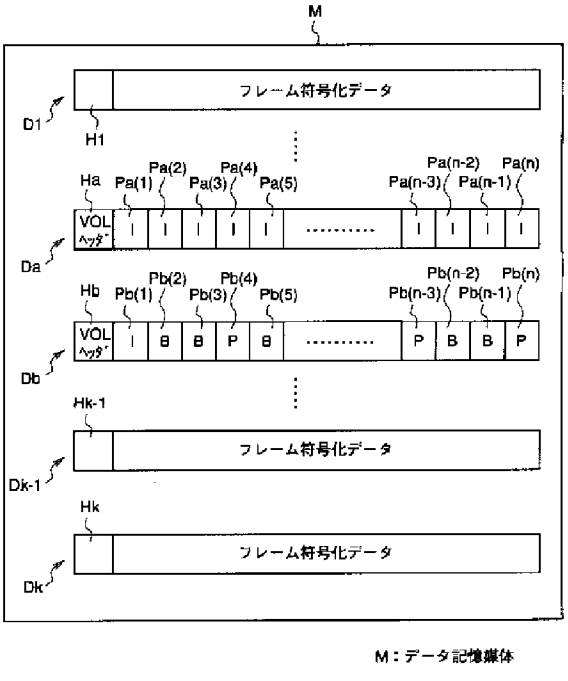

501
502
503
503a
503b
505
510
512
513
514
515
解析器
IQ
IDCT
511
523
526
524
521
525a
525b
522b
525c
522c
522a
制御器
520
517
516
予測信号
取得器
509
519
507
FM
アドレス
生成器
508
518
506
100b
100b：画像復号化装置

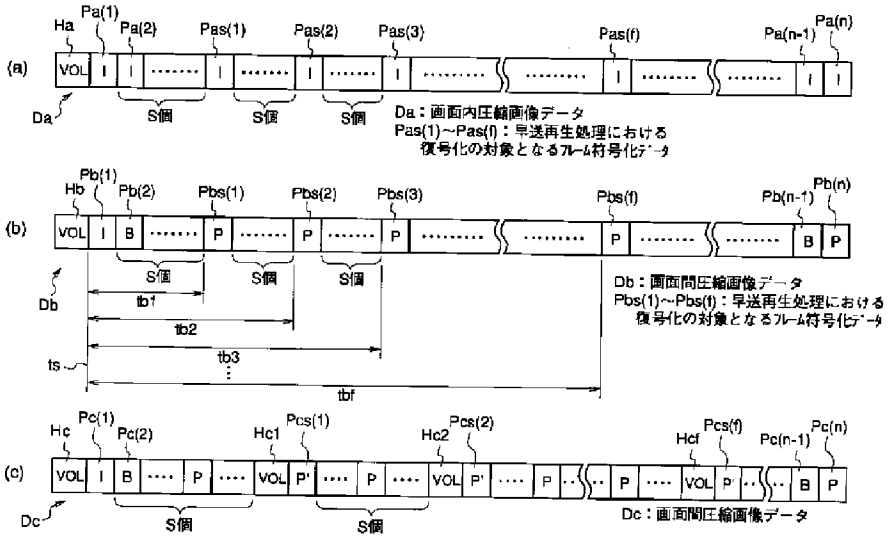

(a)
FD
FC
D
FC
FD
FC：フロッピーディスクケース
FD：フロッピーディスク
D：フロッピーディスク本体
Se：セクタ
Tr：トラック
(b)
Se
Tr
D
(c)
Cs
FDD
FD
Cs：コンピュータシステム
FDD：フロッピーディスクドライブ


(a)
ピクチャ
ピクチャ
ピクチャ
ピクチャ
ピクチャ
P(1)
P(2)
P(3)
P(n-1)
P(n)
D
D：動画データ
P(1)～P(n)：フレームデータ


(b)
Ha
Pa(1)
Pa(2)
Pa(3)
Pae(1)
Pae(2)
Pae(m)
Pa(n)
VOL
I
30秒
1時間
Da
Ha：ヘッダ
Pa(1)～Pa(n)：フレーム符号化データ
Pae(1)～Paa(m)：最後の方の30秒間に相当するフレーム符号化データ
Da：画面内圧縮画像データ
(c)
Hb
Pb(1)
Pb(2)
Pb(3)
Pb(4)
Pbe(1)
Pbe(2)
Pbe(m)
Pb(n)
VOL
I
B
P
30秒
1時間
Db
Hb：ヘッダ
Pb(1)～Pb(n)：フレーム符号化データ
Pbe(1)～Pbe(m)：最後の方の30秒間に相当するフレーム符号化データ
Db：画面間圧縮画像データ


同期信号
共通データ
RA ﾌｯｸﾞ
整列ﾃﾞｰﾀ
フレーム符号化データ
Hsd
Hcd
Hfd
Had
H
P
D
D：圧縮画像データ
H：ヘッダ


M
D1
H1
フレーム符号化データ
Ha
Pa(1)
Pa(2)
Pa(3)
Pa(4)
Pa(5)
Pa(n-3)
Pa(n-2)
Pa(n-1)
Pa(n)
VOL
ﾍｯﾀﾞ
Da
Hb
Pb(1)
Pb(2)
Pb(3)
Pb(4)
Pb(5)
Pb(n-3)
Pb(n-2)
Pb(n-1)
Pb(n)
Db
Hk-1
Dk-1
Hk
Dk
M：データ記憶媒体


(a)
Ha
Pa(1)
Pa(2)
Pas(1)
Pas(2)
Pas(3)
Pas(f)
Pa(n-1)
Pa(n)
VOL
I
Da
S個
S個
S個
Da：画面内圧縮画像データ
Pas(1)～Pas(f)：早送再生処理における
復号化の対象となるフレーム符号化データ
(b)
Hb
Pb(1)
Pb(2)
Pbs(1)
Pbs(2)
Pbs(3)
Pbs(f)
Pb(n-1)
Pb(n)
VOL
I
B
P
Db
S個
S個
S個
tb1
tb2
tb3
tbf
ts
Db：画面間圧縮画像データ
Pbs(1)～Pbs(f)：早送再生処理における
復号化の対象となるフレーム符号化データ
(c)
Hc
Pc(1)
Pc(2)
Hc1
Pcs(1)
Hc2
Pcs(2)
Hcf
Pcs(f)
Pc(n-1)
Pc(n)
VOL
I
B
P
P'
Dc
S個
S個
Dc：画面間圧縮画像データ

PICTURE STORAGE TRANSMITTER

Publication number: JP4348690 (A)
Publication date: 1992-12-03
Inventor(s): TANAKA TOMOAKI; OBA YUJI; INOMATA HIDEKI +
Applicant(s): MITSUBISHI ELECTRIC CORP; NIPPON TELEGRAPH & TELEPHONE +
Classification:
- international: ***G06T3/40; G06T9/00; H04N1/41; H04N7/173; H04N7/24; H04N7/26;*** *H04L12/28;* **G06T3/40; G06T9/00; H04N1/41; H04N7/173; H04N7/24; H04N7/26;** H04L12/28; (IPC1-7): G06F15/66; H04L12/48; H04N7/13; H04N7/173
- European:
Application number: JP19910121148 19910527
Priority number(s): JP19910121148 19910527

Also published as:
JP2609772 (B2)

Abstract of **JP 4348690 (A)**

PURPOSE:To prevent mis-recognition as cell abort when a sequence number of cell between frames is discontinuous due to the frame edit of the cell in a picture data storage section. CONSTITUTION:After each cell with a sequence number added thereto in the ascending order is stored in a picture data storage section 2, when a storage control section 3 edits each cell in the unit of frames and the sequence numbers between frames outputted one after another are discontinuous, an identification cell multiplex section 4 is activated to multiplex the identification cell between the frames. As a result, an abort discrimination section 7 recognizes the identification cell to prevent the abort of a cell in the sequence number missing part from being mis-recognized regardless of the discontinuous sequence numbers.

Data supplied from the ***espacenet*** database — Worldwide

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号

特開平4－348690

(43)公開日　平成４年(1992)12月３日

(51)Int.Cl.[5]	識別記号	庁内整理番号	Ｆ Ｉ	技術表示箇所
Ｈ０４Ｎ　7/13		Ｚ　8838－5Ｃ		
Ｇ０６Ｆ　15/66	３５５	Ａ　8420－5Ｌ		
Ｈ０４Ｎ　7/173		8324－5Ｃ		
// Ｈ０４Ｌ　12/48				
		8529－5Ｋ	Ｈ０４Ｌ　11/ 20　Ｚ	

審査請求　未請求　請求項の数１(全　５　頁)

(21)出願番号　特願平3－121148

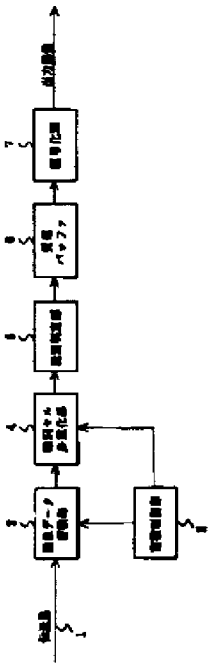
(22)出願日　平成３年(1991)５月27日

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号
(71)出願人　000004226
日本電信電話株式会社
東京都千代田区内幸町一丁目１番６号
(72)発明者　田中　知明
東京都千代田区内幸町一丁目１番６号　日本電信電話株式会社内
(72)発明者　大庭　有二
東京都千代田区内幸町一丁目１番６号　日本電信電話株式会社内
(74)代理人　弁理士　高田　守　(外２名)

最終頁に続く

(54)【発明の名称】　画像蓄積伝送装置

(57)【要約】
【目的】　この発明は、画像データ蓄積部におけるセルのフレーム編集によりフレーム間のセルのシーケンス番号が非連続になった場合に、セルの廃棄と誤認するのを防ぐことを特徴とする。
【構成】　昇順にシーケンス番号が符された各セルを画像データ蓄積部２に蓄積した後、蓄積制御部３によって各セルをフレーム編集した際、前後して出力されるフレーム間のシーケンス番号が非連続になった時、識別セル多重化部４を稼働し、フレーム間に識別セルを多重化する。この結果、廃棄判定部７では識別セルを認識することで、シーケンス番号の非連続線にも拘らず、シーケンス番号欠落部分のセルの廃棄を誤認することがない。



1

【特許請求の範囲】

【請求項1】　個々に昇順のシーケンス番号が符されたセル単位の符号化画像データを伝送路を介して蓄積部へ蓄積した後、必要に応じてフレーム編集を施して復号化部へ出力する画像蓄積伝送装置において、上記蓄積部に蓄積画像データをフレーム編集する蓄積制御部と、フレーム編集によりシーケンス番号が非連続になったフレームの前に識別セルを多重する識別セル多重化部と、前記識別セル読み取りによりセルの廃棄を判定する廃棄判定部とを備えたことを特徴とする画像蓄積伝送装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】この発明は、伝送されてきた符号化画像データを一旦蓄積した後、復号化部へ伝送する画像蓄積伝送装置に関するものである。

【０００２】

【従来の技術】以下に、本発明に関する従来技術の説明をする。図３は、テレビジョン学会誌、Ｖｏｌ．４３，Ｎｏ１０（１９８９）ｐｐ．１０７２～１０７８“周期的フレーム内符号化を用いた蓄積メディア用動画象符号化方式”および同Ｖｏｌ．４３，Ｎｏ１０（１９８９）ｐｐ．１１０６～１１１１“フレーム間符号化方式を用いた記録系動画再生システムの検討”の文中から予測した従来の画像蓄積再生装置の構成を示すものであるが、ここで３は蓄積媒体８の制御を行う蓄積制御部、９は蓄積媒体のクロックから復号クロックへ乗せ換えるための速度変換バッファ、７は符号化された画像データを可変長復号化し、さらに逆量子化を行い、モニタへ表示するための復号化部である。

【０００３】図４は、Ｂ－ＩＳＤＮの伝送方式の一種であるＡＴＭ（Ａｓｙｎｃｈｒｏｎｏｕｓ　Ｔｒａｎｓｆｅｒ　Ｍｏｄｅ：非同期転送モード）網を用いた画像伝送装置の受信部の構成を表したものであり、１は伝送路（ＡＴＭ）、５は廃棄判定部、６は伝送路でのデータの揺らぎ等を吸収するための受信バッファ、７は符号化された画像データを可変長復号し、さらに逆量子化を行い、モニタへ表示するための復号化部、１０は音声多重化されたセルの分解、合成を行うセル分解・合成部である。図５は上記ＡＴＭ網での伝送概念の一例を表したものである。各セルは昇順にシーケンス番号が符下れて蓄積媒体８に蓄積された後、蓄積制御部３によって、所定数のセル単位毎にフレーム編集される。

【０００４】次に従来技術の詳細動作について説明する。まず、従来の画像蓄積再生装置は図３のように構成されるが、ここで蓄積媒体８にはあらかじめフレーム間符号化された一定量のデータが蓄積されている。ここでは蓄積メディアとして例えば画像フレームの編集、コマ送り等ができるように、ここでの画像データは一定の周期でフレーム内符号化されている（例えば３０フレーム周期でフレーム内符号化をしていればフレーム内のみを

2

蓄積媒体から読み出せば、コマ送りとなる）。次に蓄積制御部３で蓄積媒体８からデータを読み出す制御を行う。読み出されたデータは速度変換バッファ９にてクロックの乗せ変えが行われ、復号部７で可変長復号化、逆量子化、データフォーマット変換等が行われ、モニタに出力される。

【０００５】また、従来のＡＴＭ網を用いた画像伝送装置は図４のように構成されているが、ここでＡＴＭの伝送路１から入って来た画像画像セル化データはセル分解・合成部１０で音声／映像の分解網での制御情報などが分解される。次にＡＴＭは網が輻輳した場合、優先度の低いセルが廃棄されるため、廃棄判定部５にてセルの廃棄があったかどうかを判定する。これはセル１つ１つに対して付加されているシーケンス番号の連続性を見て判定する。廃棄判定部５から出力されたデータは受信バッファ６に格納される。ここでは伝送路１でのデータ揺らぎに対する許容、また伝送クロックから復号クロックに乗せ換える機能も司る。復号化部７では可変長符号化されて伝送されて来たデータを逆に可変長復号化し、逆量子化、データフォーマットの変換等を行い、モニタに出力する。

【０００６】

【発明が解決しようとする課題】従来のＡＴＭ網用の画像伝送装置で、図３のように蓄積メディアを搭載した場合、以下のような問題点が生じる。即ち、蓄積媒体で蓄積された画像データをフレーム編集（コマ送り、逆再生等）して読み出した時、セルの順番が伝送路より入って来た時と異なってしまう。伝送路で廃棄が起こったかどうかは、先にも述べたが、セル１つ１つに対してヘッダとして付加されているシーケンス番号が連続しているかどうかで判定するので蓄積媒体でセルの順番を加工してしまうと、廃棄と誤認してしまうという問題があった。

【０００７】この発明は上記のような問題点を解決するためになされたもので、蓄積部での編集で変わってしまったシーケンス番号の連続を認識し、廃棄との区別を行い、正しい廃棄処理の行える画像蓄積伝送装置を得ることを目的とする。

【０００８】

【課題を解決するための手段】この発明に係る画像蓄積伝送装置は、個々に昇順のシーケンス番号が符されたセル単位の符号化画像データを伝送路を介して蓄積部へ蓄積した後、必要に応じてフレーム編集を施して復号化部へ出力する画像蓄積伝送装置において、上記蓄積部に蓄積画像データをフレーム編集する蓄積制御部と、フレーム編集によりシーケンス番号が非連続になったフレームの前に識別セルを多重する識別セル多重化部と、前記識別セル読み取りによりセルの廃棄を判定する廃棄判定部とを備えたものである。

【０００９】

【作用】この発明における識別セル多重化部は、制御部

Page 1759 of 2504

3

でフレーム編集シーケンス番号が非連続になったという情報（タイミング等）を受けてフレーム編集したフレームの前に識別セルを多重し、後段の廃棄判定部で編集かどうかを判断するものである。

【0010】

【実施例】以下、この発明の一実施例を図について説明する。図1において、1は網が輻輳した場合に、セルが廃棄される可能性のある伝送方式の網（例えばATM）（伝送路）、2は伝送路1より送られて来た画像データを一定量蓄積するための画像データ蓄積部、3は画像データ蓄積部2でのデータの書き込みと読み出しを制御する蓄積制御部、4は画像データ蓄積部2でフレーム編集等を行った場合に廃棄と区別するために識別セルを付加するための識別セル多重化部、5は伝送路でセルの廃棄があったかどうかを判定する廃棄判定部、6は伝送クロックから復号クロックへ乗せ換えるための受信バッファ、7は符号化された画像データを復号化しモニタへ表示するための復号化部である。

【0011】次に、この発明の一実施例について作用、動作を説明する。まず図1で、伝送路1から送られてきた画像データは、一旦画像データ蓄積部2で蓄積される。ここでの蓄積容量は画像フレームを複数フレーム蓄積できる容量を持っているものとする。この画像データ蓄積部2へデータを書き込む制御、および読み出す制御、また画像データ蓄積部2内でフレーム編集等を制御するのが蓄積制御部3であり、蓄積制御部3でフレーム編集した場合にそのフレームおよびタイミングを認識しているので、その情報を識別セル多重化部7に送る。画像データ蓄積部2から読み出されたデータ（セル）は識別セル多重化部4に入力される。ここでは、画像データ蓄積部2でフレーム編集を行わない場合、データはそのまま通過する。画像データ蓄積部2でフレーム編集を行った場合に各フレーム間のセルのシーケンス番号が非連続になった時は、蓄積制御部3で編集をしたタイミングで識別セルを多重化する。これは図2と図4に1例を表しており、それぞれの図について説明する。

【0012】まず図4であるが、これは通常のセルの並びであり、セル内の番号はパケットごと付いているシーケンス番号である。図4ではフレームごとに第1フレームから第3フレームまでを表示してあるが、ここでは編集を行って、第1フレームと第3フレームを接続した場合を述べる。図2は編集した場合のセルの並びである。第2フレームを削除してしまったために、第1フレームと第2フレームのシーケンス番号が連続でなくなっている。このため識別セルを多重しないと第2フレームは廃棄されたものとして扱われてしまう。識別セル多重化部7に手上記の処理を施した後、廃棄判定部4で廃棄されているかの判定を行い、速度変換バッファ5で伝送クロックから復号クロックに乗せ換える。そして復号部6で画像データの各種復号を行い、画像を表示するためのフォーマットに変換して出力する。

4

【0013】

【発明の効果】以上のように、この発明によれば蓄積部でのデータ編集を行った場合にセルのシーケンス番号が非連続になった時、フレーム編集を行ったフレームの前に識別セルを多重するように構成したので、セル廃棄または編集によるシーケンス番号の非連続を容易に区別でき、正しい廃棄処理が行われるという効果がある。

【図面の簡単な説明】

【図1】この発明の一実施例による画像蓄積伝送装置の構成図である。

【図2】この発明の一実施例によるセルフォーマットを示す図である。

【図3】従来の画像蓄積伝送装置の構成図である。

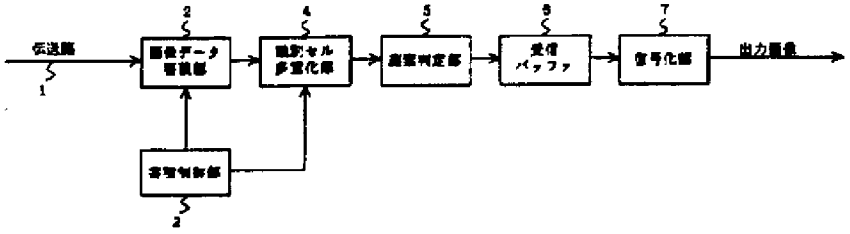
【図4】ATM用画像伝送装置の構成図である。

【図5】ATMのセルフォーマットを示す図である。

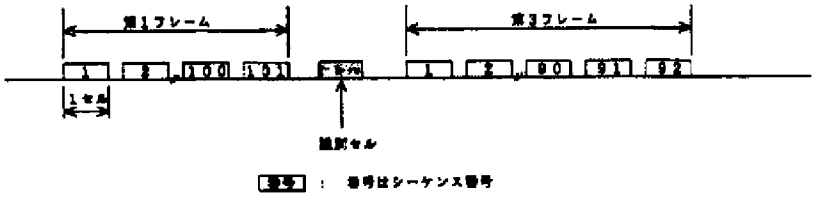
【符号の説明】

1　伝送路（ATM網）
2　画像データ蓄積部
3　蓄積制御部
4　識別セル多重化部
5　廃棄判定部
7　復号部
8　蓄積媒体
10　セル分解・合成部

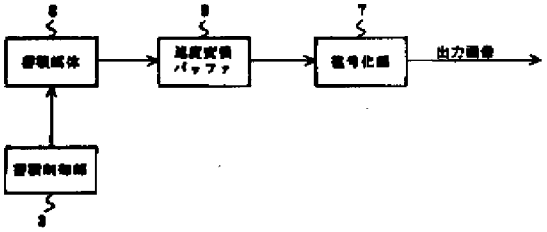
【図1】



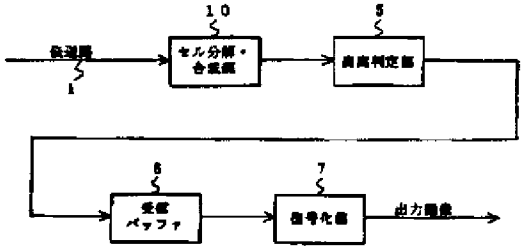
【図２】



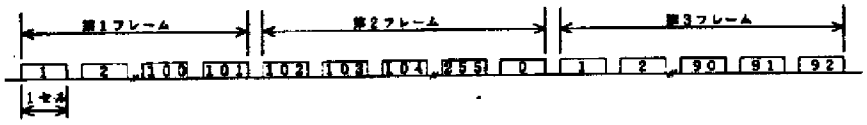
【図３】



【図４】



【図５】



【手続補正書】
【提出日】平成３年１２月１０日
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】０００３
【補正方法】変更
【補正内容】
【０００３】図４は、Ｂ－ＩＳＤＮの伝送方式の一種であるＡＴＭ（Ａｓｙｎｃｈｒｏｎｏｕｓ　Ｔｒａｎｓｆｅｒ　Ｍｏｄｅ：非同期転送モード）網を用いた画像伝送装置の受信部の構成を表したものであり、１は伝送路（ＡＴＭ）、５は廃棄判定部、６は伝送路でのデータの揺らぎ等を吸収するための受信バッファ、７は符号化された画像データを可変長復号し、さらに逆量子化を行い、モニタへ表示するための復号化部、１０は音声多重化されたセルの分解、合成を行うセル分解・合成部である。図５は上記ＡＴＭ網での伝送概念の一例を表したものである。各セルは昇順にシーケンス番号が付加されて蓄積媒体８に蓄積された後、蓄積制御部３によって、所定数のセル単位毎にフレーム編集される。

フロントページの続き

(72)発明者　猪股　英樹
神奈川県鎌倉市大船五丁目1番1号　三菱
電機株式会社通信システム研究所内

Reference Number: B065214 Dispatch Number: 582732

Mailing Date: August 17, 2010

NOTICE OF REASONS FOR REJECTION

(TRANSLATION)

Patent Application No.:	2006-120297
Drafting Date:	August 11, 2010
Patent Office Examiner:	Akiyo Sakamoto 2954 5C00
Attorney for Patent Applicant:	Atsushi Aoki (and four others)
Applicable Provisions:	Article 29, paragraph 1; Article 29, paragraph 2; Article 36

It is deemed that the present application should be rejected for the following reasons. An argument, if any, should be submitted within 3 months from the mailing date of this notice.

REASONS

a. The inventions described in the following claims of this application are inventions described in the following publications distributed or the inventions made available to the public through electronic telecommunications lines in Japan or a foreign country prior to the filing of the patent application and, therefore, are unpatentable under the provisions of Article 29, paragraph 1, item 3 of the Patent Law.

NOTE (For the numbers of the cited references, see the List of Cited References)

Claims: 1, 5, 9, 10, 13, and 14
Reference: 1
Remarks:

It is not deemed that there are any differences between the inventions according to claims 1, 5, 9, 10, 13, and 14 and the invention described in Reference 1 (refer to Reason b).

b. The inventions described in the following claims of this application are deemed ones which could easily have been made, prior to the filing of the patent application, by a person with ordinary skill in the art to which the inventions pertain, on the basis of the inventions described in the following publications distributed or the inventions made available to the public through electronic telecommunications lines in Japan or a foreign country prior to the filing of the patent application and, therefore, are unpatentable under the provisions of Article 29, paragraph 2, of the Patent Law.

NOTE (For the numbers of the cited references, see the List of Cited References)

Claims: 1, 2, 4 to 6, 8 to 11, 13, and 14
References: 1 and 2

Remarks:

Reference 1 describes that an independently reproducible flag is included in a frame header (specifically, refer to paragraphs [0112] to [0113]).

With respect to claims 2, 6, and 11, Reference 2 describes that the additional information of the frame layer is included in the header of the slice layer.

Therefore, it is deemed that including an indication identifying the image frame initially encoded in an independently decodable sequence in the header of a slice as in the present invention could have easily been conceived of by a person skilled in the art by applying the invention described in Reference 2 to the aforementioned flag of the invention described in Reference 1 to include the aforementioned flag in the header of the slice layer.

c. The description of the claims of this application does not satisfy the requirements prescribed in Article 36, paragraph 6, item 2 of the Patent Law on the following points.

Notes

There are two Japanese expressions corresponding to the "first image frame" described in claim 3, but whether or not both mean the same frame is unclear from the descriptions.

The same is true for the Japanese descriptions corresponding to "first" in other claims.

If an amendment is made, it should be noted that no new matter can be added. In an amendment, the reasons that the amendments meet the requirements prescribed in the Patent Law should be stated identifying the relevant descriptions of the original specification on which the amendments rest.

If another reason for rejection is found, a further notice of reasons for rejection will be issued.

List of Cited References

1. Japanese Unexamined Patent Publication (Kokai) No. 11-275576
2. Japanese Unexamined Patent Publication (Kokai) No. 9-187007

Record of Results of Prior Art Search

·Searched Fields IPC H04N 7/24 - 7/68

This Record of Results of Prior Art Search does not constitute a reason for rejection.

For inquiries on the content of this Notice of Reasons for Rejection or to request an interview on this case, please contact the Examiner below.

Imaging Devices Division,
Fourth Patent Examination Department
Examiner: Akiyo Sakamoto
TEL. 03(3581)1101 extension 3539
FAX. 03(3501)0715

Reference Number: B065213
Mailing Number: 595626
Mailing Date: August 24, 2010

NOTICE OF REASONS FOR REJECTION

Patent Application No.:	2006-120296
Drafting Date:	August 17, 2010
Patent Office Examiner:	Akiyo Sakamoto 2954 5C00
Attorney for Patent Applicant:	Atsushi Aoki (and four others)
Applicable Provisions:	Article 29, paragraph 2; Article 36

It is deemed that the present application should be rejected for the following reasons. An argument, if any, should be submitted within 3 months from the mailing date of this notice.

REASONS

a. The inventions described in the following claims of this application are deemed ones which could easily have been made, prior to the filing of the patent application, by a person with ordinary skill in the art to which the inventions pertain, on the basis of the inventions described in the following publications distributed or the inventions made available to the public through electronic telecommunications lines in Japan or a foreign country prior to the filing of the patent application and, therefore, are unpatentable under the provisions of Article 29, paragraph 2, of the Patent Law.

NOTE (For the numbers of the cited references, see the List of Cited References)

Claim: 26
Reference: 1
Remarks:

Reference 1 describes that when the discontinuity of the sequence number of the cells occurs between frames as a result of frame editing, the identification cells are multiplexed before the frame at which the sequence number of the cells is discontinuous, in order to distinguish the discontinuity from a discontinuity which is caused by discarding of the cells in a transmission path (specifically, refer to paragraphs [0006] to [0009]).

In the invention described in Reference 1, the discontinuity of the sequence numbers is caused by a change in the sequence of the frames as a result of frame editing, thus, it is considered that the aforementioned identification cells mean a change of the sequence of the frames by the editing.

The technology for adding a frame number to each frame is known. When the editing in the invention described in Reference 1 is performed for the image data to which the frame number is added, then, the frame number becomes discontinuous due to a change of the sequence of the frames, thus, the discontinuity of the frame number is represented by the aforementioned identification cell.

Therefore, it is deemed that informing that a discontinuity in the numbering of the image frames is intentional as in the invention of the present application,

could have easily been conceived of by a person skilled in the art based on the invention described in Reference 1 and the aforementioned known technology.

b. The description of the claims of this application does not satisfy the requirements prescribed in Article 36, paragraph 6, item 2 of the Patent Law on the following points.

Notes

(1) The Japanese description corresponding to "the video sequence in question" is described in claims 14 and 22, but from the description of "the...in question", what is indicated by the description of "the" is unclear.

(2) "a computer program comprising steps to be executed by a computer...step of (omitted)..., step of...(omitted)...." is described in claim 18, but from the description of "to be executed by a computer...step of (omitted)...", whether or not the computer program causes a computer to execute each step is unclear(refer to Part VII, Chapter 1, 1.1.1(2)(b) of Examination Guidelines for Patent and Utility Model).

Note that, the above reason is overcome if the description of "a computer program comprising steps to be executed by a computer, the steps including the step of decoding a compressed video sequence, a step of...(omitted)...." is amended to, for example, "a computer program for causing a computer to execute a step of decoding a compressed video sequence, a step of...(omitted)....".

(3) Claims 22 to 25 describe "...comprises code for...", but

claim 18 upon which claims 22 to 25 are dependent upon uses not the term "code", but uses the term "step", thus, claims 22 to 25 which describe "comprises code" contradict the description of claim 18 and are unclear.

Note that, the above reason is overcome if the term "code" of claims 22 to 25 is replaced with, for example, "a step".

Further, if an amendment is made, it should be noted that no new matter can be added and in an argument, the reasons that the amendments meet the requirements of the Patent Law should be stated, identifying the relevant descriptions of the original specification on which the amendments are based.

<Claims for which no reasons for rejection have been found>

No reasons for rejection have been found at present for the inventions according to claims 1 to 13. If another reason for rejection is found, a further notice of reasons for rejection will be issued.

List of Cited References

1. Japanese Unexamined Patent Publication (Kokai) No. 4-348690

Record of Results of Prior Art Search

·Searched Fields IPC
H04N 5/76 - 5/765, H04N 5/80 - 5/956, H04N 7/10 - 7/173, H04N 7/20 - 7/68, H04L1/00, H04L 1/08 - 1/24

This Record of Results of Prior Art Search does not constitute a reason for rejection.

For inquiries on the content of this Notice of Reasons for Rejection or a request for an interview on this case, please contact the Examiner below.

Akiyo Sakamoto, Examiner
Imaging Devices, Fourth Examination Division
TEL. 03 (3581) 1101 extension 3539
FAX. 03 (3501) 0715

Electronic Acknowledgement Receipt

EFS ID:	8625005
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Steven R. Funk
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	14-OCT-2010
Filing Date:	25-JAN-2006
Time Stamp:	14:34:15
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-12-TPO004C1.pdf	57769 bee87e113f8d5734f94c7060b337ed140fba9d08	no	2

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	Form1449-12-TPO004C1.pdf	71881 d24ca9c6d26f69c74a54edf1220f25e42db3c67c	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Foreign Reference	JP11275576pre.pdf	236628 f872f28d2862519b5f1d6ad930ba7863892ec8e3	no	35
Warnings:					
Information:					
4	Foreign Reference	JP4348690pre.pdf	188895 cabd1429d49abfe3242ac88d82326510350983ea	no	6
Warnings:					
Information:					
5	NPL Documents	JPOA20100817.pdf	74168 f318ac7a8fbcd4fe55a3d0fd18432cfe25dd0936	no	4
Warnings:					
Information:					
6	NPL Documents	JPOA20100824.pdf	94278 a850e4ac8e953ae55e3060329b123b4f4e6e0fda	no	5
Warnings:					
Information:					
		Total Files Size (in bytes):	723619		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. §1.97(c)(1) and 37 C.F.R. §1.97 (e)(1)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted after the mailing date of the first Office Action on the merits but before the mailing date of a final Office Action under 37 C.F.R. §1.113 or a Notice of Allowance under 37 C.F.R. §1.311.

The items listed on the enclosed Form 1449 were first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449. However, if fees are deemed necessary, authorization is given to charge Deposit Account 50-3581 (TPO.004.C1) for consideration of the information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. § 1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Steven R. Funk/
Steven R. Funk
Reg. No. 37,830

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,996	01/25/2006	Miska Hannuksela	TPO.004.C1	5361

10888 7590 12/09/2010

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

MAIL DATE	DELIVERY MODE
12/09/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/338,996	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	SIU M. LEE	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 September 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6,10-15 and 18-23* is/are rejected.

7)☒ Claim(s) *7,8,16,17,24 and 25* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 January 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☒ Certified copies of the priority documents have been received in Application No. 10/306,942.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *See Continuation Sheet*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :8/10/2010, 9/30/2010, 10/14/2010.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments, see page 8, filed 9/10/2010, with respect to claim objection to claims 1, 5, 9, 10, 14, 18, 22, and 24 and objection to figure 1 have been fully considered and are persuasive. The objection of claims 1, 5, 9, 10, 14, 18, 22, and 24 and objection to figure 1 have been withdrawn.

2. Applicant's arguments, see page 8, filed 9/10/2010, with respect to 35 U.S.C. 101 rejection to claims 18-25 have been fully considered and are persuasive. The 35 U.S.C. 101 rejection to claims 18-25 has been withdrawn.

3. Applicant's arguments filed 9/10/2010 with respect to the 35 U.S.C. 101 rejection of claims 10-17 have been fully considered but they are not persuasive.

Applicant's argument:

The claimed video decoder is a device constituting statutory subject matter. The term "decoder" is well known in the art to refer to a device or circuit. Notably, the specification illustrates structural component of the decoder, and independent claim 10 recites that the video decoder is connected to a buffer memory. Paragraph 006 describes positioning of the decoder that would not be possible if it was "pure software".

Examiner's response:

Page 33, paragraph 0097 of the specification of the instant application clearly disclose that the decoder can be implemented by software, hardware, or a combination of hardware and software:

"**[0097] It is to be noted that according to the invention the functional elements of the above described streaming system and its elements, such as the streaming server, video encoder, video decoder and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two.**"

Base on the specification and the fact that the claimed decoder is non-structural per se, the claimed video decoder covers hardware, software, and a combination of hardware and software embodiment. In light of the specification, a broadest reasonable interpretation of the claimed video decoder can be interpreted as software per se. Therefore, the 35 U.S.C. 101 rejection of claims 10-17 are maintained.

4. Applicant's arguments filed 9/10/2010 with respect to the obviousness-type double patenting rejection of claims 1-6, 10-15, and 18-23 over co-pending and commonly owned patent application 10/306,942, the applicant states that the applicant will consider filing a terminal disclaimer once the instant application is in condition for allowance or the application 10/306942 application issues as a patent.

According to the record of the application 10/306,942, the application has been allow on 10/7/2010; therefore, the examiner maintain the double patenting rejection on

claims 1-6, 10-15, and 18-23. The applicant can file a terminal disclaimer to overcome the double patenting rejection on claims1-6, 10-15, and 18-23.

Claim Objections

5. Claim 10-17 are objected to because of the following informalities:

Line 4 recites “decode from the decompressed video sequence”; there is a lack of antecedent basis for “the decompressed video sequence”. The examiner suggests changing to “decode from the **compressed** video sequence".

Appropriate correction is required.

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 10-17 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claimed “video decoder” is non-structural per se and because the specification discloses a “software” embodiment (page 6, paragraph 0015 and page 33, paragraph 0097). Therefore, a reasonable interpretation in light of the specification leads to the conclusion that the claim video decoder covers a software embodiment, which does not fall within the definition of a process, machine, manufacture or composition of matter.

Double Patenting

7. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

1. Claims 1-5 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 17 of copending Application No. 10/306942.

This is a provisional obviousness-type double patenting rejection.

Claim	Instant application	Claim	Co-pending application 10/306942

1	A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising: decoding from the compressed video sequence an indication related to at least one discontinuity of numbering of the image frames; configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames; and using the image frames in the buffer memory in the decoding process.	17	(including limitation of claim 16) Claim 16 A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, the method comprising: decoding, at a decoder, a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format; decoding, at the decoder, at least a second sub-sequence of the

10	A video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding, the video decoder being configured to decode from the compressed video sequence an indication related to at least one discontinuity of numbering of the image frames; configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames; and use the image frames in the buffer memory in the decoding process.		video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; determining, at the decoder, dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and reconstructing, at the decoder, at least part of the video sequence on the basis of the dependencies of the sub-sequences. Claim 17 A method according to claim 16, **wherein video frames are entered into a sliding buffer**

			memory in connection with decoding, further comprising: **decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional**; **configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers**; and continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to

			the missing image numbers.

(1) Regarding claim 1:

Claim 17 of the copending application discloses decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames; and configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to a discontinuity in the numbering of the image frames (the bold and underlined part in the above table).

The co-pending application does not explicitly disclose using the image frame in the buffer memory in the decoding process.

However, claim 17 also discloses wherein video frames are entered into a sliding buffer memory in connection with decoding (preamble of claim 17).

Therefore, it is obvious that the image frame in the buffer memory in the decoding process for the benefit of performing decoding with a continuous frame sequence number.

(2) Regarding claim 2:

Claim 17 further disclose wherein said indication informs that at least one discontinuity in the numbering of the image frames in the video sequence are intentional (the discontinuity in the numbering of the images of the image frames in the video sequence are intentional, claim 17).

(3) Regarding claim 3:

The co-pending application does not explicitly discloses using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

However, claim 17 discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

As the video frames are enter into a sliding buffer memory in connection with decoding, it is obvious that the comprise number of image frames corresponding to the missing image number are used in the decoding process in place of the missing frame number to provide a benefit of performing decoding with a continuous sequence number.

(4) Regarding claim 4:

Claim 17 further disclose the number of image frames in the buffer memory are used in sliding window buffering of image frame (wherein video frames are entered into a sliding buffer memory in connection with decoding).

(5) Regarding claim 5:

Clam 17 further discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames (continuing the entering of the image frames comprised by the video sequence in

question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers, claim 17).

2. Claim 6 is provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 18 of copending Application No. 10/306942.

This is a provisional obviousness-type double patenting rejection.

Claim 17 discloses all subject matter as discussed in above and claim 18 of the co-pending application depends on claim 17 and further disclose entering onto the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames (entering into the buffer memory a number of filler frames corresponding to the missing image number, claim 18).

3. Claims 10-14 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 17 of copending Application No. 10/306942.

(1) Regarding claim 10:

Claim 17 of the copending application discloses decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames; and configuring, in response to the indication, the buffer memory to provide a

number of image frames corresponding to a discontinuity in the numbering of the image frames (the bold and underlined part in the above table).

The co-pending application does not explicitly disclose (a) using the image frame in the buffer memory in the decoding process; and (b) the decoding method is performed by a decoder.

With respect to (a), claim 17 also discloses wherein video frames are entered into a sliding buffer memory in connection with decoding (preamble of claim 17).

Therefore, it is obvious that the image frame in the buffer memory in the decoding process for the benefit of performing decoding with a continuous frame sequence number.

With respect to (b), since the claim is non-structural per se, and claim 17 discloses the decoding steps, it is inherently that the decoding method is performed by a decoder.

(2) Regarding claim 11:

Claim 17 further disclose wherein said indication informs that at least one discontinuity in the numbering of the image frames in the video sequence are intentional (the discontinuity in the numbering of the images of the image frames in the video sequence are intentional, claim 17).

(3) Regarding claim 12:

The co-pending application does not explicitly discloses using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

However, claim 17 discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

As the video frames are enter into a sliding buffer memory in connection with decoding, it is obvious that the comprise number of image frames corresponding to the missing image number are used in the decoding process in place of the missing frame number to provide a benefit of performing decoding with a continuous sequence number.

(4) Regarding claim 13:

Claim 17 further disclose the number of image frames in the buffer memory are used in sliding window buffering of image frame (wherein video frames are entered into a sliding buffer memory in connection with decoding).

(5) Regarding claim 14:

Clam 17 further discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames (continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers, claim 17).

4. Claim 15 is provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 18 of copending Application No. 10/306942.

This is a provisional obviousness-type double patenting rejection.

Claim 17 of the co-pending application discloses all subject matter as discussed in above and claim 18 of the co-pending application depends on claim 17 and further disclose entering onto the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames (entering into the buffer memory a number of filler frames corresponding to the missing image number, claim 18, since the claim is non-structural per se, and claim 17 discloses the decoding steps, it is inherently that the decoding method is performed by a decoder).

5. Claims 18-22 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 17 of copending Application No. 10/306942 in view of Langberg et al. (US 5,852,630, hereinafter Langberg).

(1) Regarding claim 18:

Claim 17 of the copending application discloses decoding from the video sequence an indication related to at least one discontinuity of numbering of the image frames; and configuring, in response to the indication, the buffer memory to provide a

number of image frames corresponding to a discontinuity in the numbering of the image frames (the bold and underlined part in the above table).

The co-pending application does not explicitly disclose (a) using the image frame in the buffer memory in the decoding process; and (b) the decoding method is perform by a computer program product stored on a computer readable medium and having program code to perform the method.

With respect to (a), claim 17 also discloses wherein video frames are entered into a sliding buffer memory in connection with decoding (preamble of claim 17).

Therefore, it is obvious that the image frame in the buffer memory in the decoding process for the benefit of performing decoding with a continuous frame sequence number.

With respect to (b), Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of claim 17 of the co-pending application would have been implemented in software. The implemented software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software in as taught by Langberg to implement the method of the co-

pending application in order to reduce cost and improve the adaptability and flexibility of the communication system.

(2) Regarding claim 19:

Claim 17 further disclose wherein said indication informs that at least one discontinuity in the numbering of the image frames in the video sequence are intentional (the discontinuity in the numbering of the images of the image frames in the video sequence are intentional, claim 17).

(3) Regarding claim 20:

The co-pending application does not explicitly discloses using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the discontinuities in the numbering of the image frames.

However, claim 17 discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

As the video frames are enter into a sliding buffer memory in connection with decoding, it is obvious that the comprise number of image frames corresponding to the missing image number are used in the decoding process in place of the missing frame number to provide a benefit of performing decoding with a continuous sequence number.

(4) Regarding claim 21:

Claim 17 further disclose the number of image frames in the buffer memory are used in sliding window buffering of image frame (wherein video frames are entered into a sliding buffer memory in connection with decoding).

(5) Regarding claim 22:

Clam 17 further discloses continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the discontinuities in the numbering of the image frames (continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers, claim 17).

6. Claim 23 is provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 18 of copending Application No. 10/306942.

This is a provisional obviousness-type double patenting rejection.

Claim 17 and Langberg discloses all subject matter as discussed in above and Claim 18 of the co-pending application depends on claim 17 and disclose entering onto the buffer memory a number of filler frames corresponding to the discontinuities in the numbering of the image frames (entering into the buffer memory a number of filler frames corresponding to the missing image number, claim 18, since the claim is non-

structural per se, and claim 17 discloses the decoding steps, it is inherently that the decoding method is performed by a decoder).

Allowable Subject Matter

Claims 7-8, 16-17, and 24-25 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU M. LEE whose telephone number is (571)270-1083.

The examiner can normally be reached on Mon-Fri, 7:30-4:00 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Examiner, Art Unit 2611
11/1/2010

/CHIEH M FAN/
Supervisory Patent Examiner, Art Unit 2611

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	9/17/2010	Office Action Response dated 9/17/2010 from U.S. Serial No. 11/338,934, 10 pages.
/SL/	9/15/2010	Office Action Response dated 9/15/2010 from U.S. Serial No. 10/306,942, 12 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 10/30/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/SL/	4390966	06/28/1983	Kawashima et al.			
/SL/	5852630	12/22/1998	Langberg			
/SL/	6072831	06/06/2000	Chen			
/SL/	6094456	07/25/2000	Ueda			
/SL/	6363208	03/26/2002	Nitta et al.			
/SL/	20010040700	11/15/2001	Hannuksela et al.			
/SL/	20030063806	04/03/2003	Kim et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	2137197C1	9/10/1999	Russia				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	1997	Illgner et al., "Spatially Scalable Video Compression Employing Resolution Pyramids", IEEE Journal on Selected Areas in Communications, Vol. 15, No. 9, December 1, 2007, pg. 1688-1703.
/SL/	2000	Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Systems, ISO/IEC 13818-1, Second edition, December 1, 2000, 174 pages.
/SL/	5/5/2010	Notice of Allowance dated 5/5/2010 from Russian Application No. 2006110321/09, 11 pages.
/SL/	5/25/2010	Office Action dated 5/25/2010 from U.S. Serial No. 11/338,934, 29 pages.
/SL/	6/23/2010	Office Action dated 6/23/2010 from U.S. Serial No. 10/306,942, 6 pages.
/SL/	5/25/2010	Office Action dated 5/25/2010 from U.S. Serial No. 10/306,942, 6 pages.
/SL/	4/09/2009	Office Action dated 4/9/2009 from U.S. Serial No. 10/306,942, 4 pages.
/SL/	5/8/2006	Office Action Response dated 5/8/2006 from U.S. Serial No. 10/306,942, 11 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 10/30/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	06/13/2010	11/01/2010							
	1	✓	✓							
	2	✓	✓							
	3	✓	✓							
	4	✓	✓							
	5	✓	✓							
	6	✓	✓							
	7	O	O							
	8	O	O							
	9	O	✓							
	10	✓	✓							
	11	✓	✓							
	12	✓	✓							
	13	✓	✓							
	14	✓	✓							
	15	✓	✓							
	16	✓	O							
	17	✓	O							
	18	✓	✓							
	19	✓	✓							
	20	✓	✓							
	21	✓	✓							
	22	✓	✓							
	23	✓	✓							
	24	✓	O							
	25	✓	O							
	26	✓	-							
	27	✓	-							
	28	✓	-							
	29	✓	-							
	30	✓	-							
	31	✓	-							

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

SEARCHED

Class	Subclass	Date	Examiner
375	240.27, 240.25	6/14/2010	Siu M. Lee

SEARCH NOTES

Search Notes	Date	Examiner
EAST	6/14/2010	Siu M. Lee
Discuss the application with Chieh Fan	6/14/2010	Siu M. Lee
EAST	11/1/2010	Siu M. Lee
Consult John Peng on the 101 problem in claim 10-17	11/1/2010	Siu M. Lee
Discuss the amendment with Cheih Fan	10/27/2010	Siu M. Lee

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	JP11275576	10/08/1999	JP			Abstract only	
/SL/	JP4348690	12/03/1992	JP			Abstract only	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	08/17/2010	Copy of translated Office Action dated 08/17/2010 from parallel Japanese Application No. 2006-120297, 4 pages.
/SL/	08/24/2010	Copy of translated Office Action dated 08/24/2010 from parallel Japanese Application No. 2006-120296, 5 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 10/30/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 5361

SERIAL NUMBER	FILING or 371(c) DATE / RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/338,996	01/25/2006	375	2611	TPO.004.C1

APPLICANTS
Miska Hannuksela, Tampere, FINLAND;

**** CONTINUING DATA ***************************
This application is a CON of 10/306,942 11/29/2002

**** FOREIGN APPLICATIONS ***************************
FINLAND 20020127 01/23/2002

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****
02/28/2006

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	Met after Allowance	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
☑ Yes ☐ No	☑ Yes ☐ No	/SIU M LEE/ Examiner's Signature	☐ Initials	FINLAND	4	31	6

ADDRESS

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

TITLE

Grouping of image frames in video coding

FILING FEE RECEIVED		
2150	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

BIB (Rev. 05/07).

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	3	"4390966".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:39
L2	2	"5852630".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:39
L3	2	"6072831".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:40
L4	2	"6094456".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:42
L5	2	"6363208".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:44
L6	2	"20100040700".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:45
L7	2	"20010040700".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:46

L8	2	"20030063806".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:47
S1	3	10/306942	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/02 13:46
S2	1	11/338996 and software	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/02 13:58
S3	1	11/338996 and (computer readable medium)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 07:09
S4	2	"5122875".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 08:05
S5	2	"5838265".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 08:08
S6	3	10/306942	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 08:20
S7	2	10/306942 and (decod$3 with buffer with memory with discontinuity)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 08:26
S8	2	"5877812".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 09:15

S9	3	10/306942 and (buffer with memory)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 09:27
S10	2	10/306942 and (buffer with memory with discontinuity)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 09:39
S11	2	10/306942 and (buffer with memory with discontinuity) and discontinuity	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 09:39
S12	2	10/306942 and decod$3 and (buffer with memory with discontinuity) and discontinuity	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 10:26
S13	49	(null or dummy or filler) with frame with (memory or buffer or storage or register) with decod $3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 14:27
S14	30	S13 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 14:28
S15	581	(frame with (identifier or indicator or tag or number$3)) with discontinu$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 15:04
S16	368	S15 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 15:04
S17	76	S16 and ((buffer or memory or register or stor$3) with decoding)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 15:05

S19	31	S17 and video	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/03 15:09
S20	20	number$3 with ((video or image) frame) with discontinu$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:34
S21	0	discard$3 with (image or video) frame with discontinu$4 with number$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:55
S22	0	discard$3 with (image or video) frame with discontinu$4 with label$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:55
S23	0	discard$3 with (image or video) frame with discontinu$4 with tag	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:55
S24	62	discard$3 with (image or video) frame with number$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:55
S25	33	S24 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/04 11:56
S26	2636	375/240.27,240.25.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/13 20:57
S27	1417	S26 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/13 20:57

S28	49	S27 and ((discontinu$4 or miss $3 or skip$4) with frame with number$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/06/13 20:59
S29	2	"6363208".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/10/30 14:51
S30	47	(indicat$3 or number$3) with ((video or image) frame) with discontinu$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:22
S31	25	S30 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:22
S32	8	S31 and ((buffer or memory) with discontinu$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2010/11/02 08:25

EAST Search History (Interference)

<This search history is empty>

11/2/2010 8:49:50 AM
C:\Documents and Settings\slee7\My Documents\EAST\Workspaces\11338996 201006.wsp

PTO/SB/25 (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TERMINAL DISCLAIMER TO OBVIATE A PROVISIONAL DOUBLE PATENTING REJECTION OVER A PENDING "REFERENCE" APPLICATION	Docket Number (Optional) TPO.004.C1

In re Application of: Hannuksela

Application No.: 11/338,996

Filed: January 25, 2006

For: Grouping of Image Frames in Video Coding

The owner*, Nokia Corporation, of 100 percent interest in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of any patent granted on pending **reference** Application Number 10/306,942, filed on November 29, 2002, as such term is defined in 35 U.S.C. 154 and 173, and as the term of any patent granted on said **reference** application may be shortened by any terminal disclaimer filed prior to the grant of any patent on the pending **reference** application. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and any patent granted on the **reference** application are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of any patent granted on the instant application that would extend to the expiration date of the full statutory term as defined in 35 U.S.C. 154 and 173 of any patent granted on said **reference** application, "as the term of any patent granted on said **reference** application may be shortened by any terminal disclaimer filed prior to the grant of any patent on the pending **reference** application," in the event that: any such patent: granted on the pending **reference** application: expires for failure to pay a maintenance fee, is held unenforceable, is found invalid by a court of competent jurisdiction, is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321, has all claims canceled by a reexamination certificate, is reissued, or is in any manner terminated prior to the expiration of its full statutory term as shortened by any terminal disclaimer filed prior to its grant.

Check either box 1 or 2 below, if appropriate.

1. ☐ For submissions on behalf of a business/organization (e.g., corporation, partnership, university, government agency, etc.), the undersigned is empowered to act on behalf of the business/organization.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

2. ☑ The undersigned is an attorney or agent of record. Reg. No. 57,125

/Erin Nichols Matkaiti/ — Signature
January 31, 2011 — Date

Erin Nichols Matkaiti — Typed or printed name

952-232-6907 — Telephone Number

☑ Terminal disclaimer fee under 37 CFR 1.20(d) is included.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this statement. See MPEP § 324.

This collection of information is required by 37 CFR 1.321. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on January 31, 2011.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.116

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action dated December 9, 2010, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.116.

Claims 1-25 remain pending in the application. Claim 10 has been amended. No claims have been added or canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE SPECIFICATION begin on page 2.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 3 of this paper.

REMARKS/ARGUMENTS begin on page 9.

AMENDMENTS TO THE SPECIFICATION

Please amend the paragraph at page 33, lines 6-16, as indicated below.

[0097] It is to be noted that according to the invention the functional elements of the above described streaming system and its elements, such as the streaming server, video encoder, video decoder and terminal are preferably implemented by ~~means of software, by~~ hardware solutions[[,]] or as a combination of ~~the two~~hardware and software. The coding and decoding methods of the invention are particularly suitable for implementation as computer software comprising computer-readable commands for executing the process steps of the invention. A preferred way of implementing the encoder and the decoder is to store them in a storage means as a program code that can be executed by a computer-like device, for example a personal computer (PC) or a mobile station, to provide coding/decoding functionalities on the device in question.

AMENDMENTS TO THE CLAIMS

Applicant respectfully requests that the amendments presented below be entered and that this listing of claims replace all prior versions of claims in the application.

1. (Previously presented) A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising:

decoding from the compressed video sequence an indication related to at least one discontinuity of numbering of the image frames;

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames; and

using the image frames in the buffer memory in the decoding process.

2. (Previously presented) A method according to claim 1, wherein said indication informs that the at least one discontinuity in the numbering of the image frames in the video sequence is intentional.

3. (Previously presented) A method according to claim 1, further comprising

using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

4. (Original) A method according to claim 1, wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.

5. (Previously presented) A method according to claim 1, further comprising

continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer

memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

6. (Previously presented) A method according to claim 1, further comprising

entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

7. (Previously presented) A method according to claim 6, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.

8. (Previously presented) A method according to claim 1, wherein

configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.

9. (Previously presented) A method according to claim 3, further comprising

detecting an error in the compressed video sequence on a basis of a reference in the decoding process to an image frame corresponding to the at least one discontinuity in the numbering of image frames.

10. (Currently amended) A video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding, the video decoder being configured to

decode from the ~~decompressed~~compressed video sequence an indication related to at least one discontinuity of numbering of the image frames;

configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames; and

use the image frames in the buffer memory in the decoding process.

11. (Previously presented) A video decoder according to claim 10, wherein

said indication informs that the at least one discontinuity in the numbering of the image frames in the video sequence is intentional.

12. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

use the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

13. (Original) A video decoder according to claim 10, wherein

the number of image frames in the buffer memory are used in sliding window buffering of image frames.

14. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

continue the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

15. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

enter into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

16. (Previously presented) A video decoder according to claim 15, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.

17. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

configure the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

18. (Previously presented) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, image frames of said video sequence being entered into a buffer memory in connection with the decoding, the computer program product comprising:

a computer program code for decoding from the compressed video sequence an indication related to at least one discontinuity of numbering of the image frames;

a computer program code for configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames; and

a computer program code for including the image frames in the buffer memory in the decoding process.

19. (Previously presented) A computer program product according to claim 18, wherein said indication informs that the at least one discontinuity in the numbering of the image frames in the video sequence is intentional.

20. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for including the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

21. (Original) A computer program product according to claim 18, the computer program product further comprising

a computer program code for using the number of image frames in the buffer memory in sliding window buffering of image frames.

22. (Previously presented) A computer program product according to claim 18, the computer program product further comprising

a computer program code for continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

23. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

24. (Previously presented) A computer program product according to claim 23, the computer program product further comprising:

a computer program code for designating said filler frames by an identifier indicating that the filler frames do not belong to the compressed video sequence.

25. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

26-31. (Canceled)

REMARKS/ARGUMENTS

The rejections presented in the Office action dated December 9, 2010 (hereinafter Office action) have been considered, and reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

Applicant appreciates the indication of allowability for claims 7, 8, 16, 17, 24, and 25.

With respect to the objection to claims 10-17, claim 10 has been amended to correct the typographical error in accordance with the Examiner's suggestion. Since this change is believed to overcome the objection, Applicant requests that the objection be removed.

With respect to the § 101 rejection of claims 10-17, Applicant respectfully maintains that the claimed decoder is directed to statutory subject matter. While the specification refers to embodiments involving computer readable media storing program code (*e.g.*, claim 18), the claimed video decoder is a device constituting statutory subject matter. The cited portion at page 6, paragraph [0015] makes no mention of a software only embodiment. The term "decoder" is well-known in the art to refer to a device or circuit. Notably, the specification illustrates structural components of the decoder (*e.g.*, demultiplexer 116 and decompressor 118), and independent claim 10 recites that the video decoder is connected to a buffer memory. Paragraph [0095] teaches examples of a decoder including a low bit rate video decoder as described in the ITU-T recommendation H.263 or H.26L. Moreover, paragraph [0096] describes positioning of the decoder that would not be possible if it was only software. However, paragraph [0097] has been amended to remove an interpretation of the claimed decoder as being software only. Since the claimed decoder is directed to statutory subject matter, Applicant requests that the rejection be withdrawn.

With respect to the provisional double patenting rejections, each of which is based on co-pending application serial number 10/306,942, Applicant submits the attached Terminal Disclaimer. Applicant accordingly requests that each of the provisional non-statutory double patenting rejections be withdrawn.

In view of the above, Applicant believes each of the pending claims is in condition for allowance and requests that the application be allowed.

It should also be noted that Applicant does not acquiesce to the Examiner's statements or conclusions concerning what would have been obvious to one of ordinary skill in the art, obvious design choices, inherent, common knowledge at the time of Applicant's invention, officially noticed facts, and the like. Applicant reserves the right to address in detail the Examiner's characterizations, conclusions, and rejections in future prosecution.

Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C1) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the undersigned attorney of record invites the Examiner to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.232.6907

Date: January 31, 2011

By: /Erin Nichols Matkaiti/

Erin Nichols Matkaiti
Reg. No. 57,125

Electronic Patent Application Fee Transmittal

Application Number:	11338996
Filing Date:	25-Jan-2006
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Filer:	Erin M. Nichols
Attorney Docket Number:	TPO.004.C1

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Statutory or terminal disclaimer	1814	1	140	140
		Total in USD ($)		**140**

Electronic Acknowledgement Receipt

EFS ID:	9342760
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	10888
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	31-JAN-2011
Filing Date:	25-JAN-2006
Time Stamp:	15:15:44
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$140
RAM confirmation Number	1816
Deposit Account	503581
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Terminal Disclaimer Filed	TermDisc-TPO004C1-OA09122010.pdf	207366 670d81bc51236af3b731458e88c6f9c398cfdbf4	no	2
Warnings:					
Information:					
2		OAR-TPO004C1-OA09122010.pdf	87342 109727e2ee4724066077f31933d23915c495264b	yes	10

Multipart Description/PDF files in .zip description

Document Description	Start	End
Amendment After Final	1	1
Specification	2	2
Claims	3	8
Applicant Arguments/Remarks Made in an Amendment	9	10

Warnings:					
Information:					
3	Fee Worksheet (PTO-875)	fee-info.pdf	29929 e4ccb3b70e2559a62775fd30adaea259168ab16b	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	324637		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 11/338,996	Filing Date 01/25/2006	☐ To be Mailed

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY ☐		OR	OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)		RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	01/31/2011	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 25	Minus	** 31	= 0	X $ =		OR	X $52=	0
	Independent (37 CFR 1.16(h))	* 6	Minus	***6	= 0	X $ =		OR	X $220=	0
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/LASHAWN MARKS/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,996	01/25/2006	Miska Hannuksela	TPO.004.C1	5361

10888 7590 02/18/2011

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

NOTIFICATION DATE	DELIVERY MODE
02/18/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

tdotter@hfiplaw.com
roswood@hfiplaw.com

Office Action Summary	Application No.	Applicant(s)	
	11/338,996	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	SIU M. LEE	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 January 2011*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-10,12-18 and 20-25* is/are rejected.

7)☒ Claim(s) *2,11 and 19* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 January 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☒ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Transitional After Final Practice

1. Since this application is eligible for the transitional procedure of 37 CFR 1.129(a), and the fee set forth in 37 CFR 1.17(r) has been timely paid, the finality of the previous Office action is hereby withdrawn pursuant to 37 CFR 1.129(a). Applicant's first submission after final filed on 1/31/2011 has been entered.

Response to Arguments

2. Applicant's arguments, see page 9-10, filed 1/31/2011, with respect to Double Patenting rejection of claims 1-6, 10-15, 18-23 and 35 U.S.C. 101 rejection of claims 10-17 have been fully considered and are persuasive. The Double Patenting rejection of claims 1-6, 10-15, 18-23 and 35 U.S.C. 101 rejection of claims 10-17 have been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Apostolopoulos (US 7,103,669 B2).

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States

only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1 are rejected under 35 U.S.C. 102(e) as being anticipated by Apostolopoulos (US 7,103,669 B2).

(1) Regarding claim 1:

Apostolopoulos disclose a method for decoding a compressed video sequence (odd number sequence ort even number sequence), wherein image frames are entered into a buffer memory in connection with the decoding (decode 322 in figure 5 enter the decoded frame into the frame memory including the previous decoded frames as shown in figure 5), the method comprising:

decoding from the compressed video sequence an indication related to at least one discontinuity of numbering of the image frame (the packet is decoded in step 914 and a determination is made whether an error has been detected, when an error is detected, state recovery is performed by employing previous and future frames of correctly decoded frames, column 7, lines 9-37; the examiner interprets the claimed discontinuity of numbering as the detected error that one of the frame (for example P5 as discussed in column 10, lines 52-63) is unable to be retrieved due to packet loss);

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to the art least one discontinuity in the numbering of the image frames (the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18); and

using the image frames in the buffer memory in the decoding process (the examiner interprets the process within the dot line as shown in figure 9 as the decoding process, as the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18, therefore, full frame versions of P2, P4, and P6 through P14, as well we P1 and P3 in the buffer memory are use in the decoding process).

(2) Regarding claim 10:

Apostolopoulos discloses a video decoder (multiple state video decoding 128 in figure 1 comprises decode block 320 and 322 as shown in figure 5) for decoding a compressed video sequence, the video decoder being connected to a buffer memory (frame memory as shown in figure 5) wherein image frames are entered in connection with the decoding, the video decoder being configured to

decode from the compressed video sequence an indication related to at least one discontinuity of numbering of the image frames (the packet is decoded in step 914 and a determination is made whether an error has been detected, when an error is detected, state recovery is performed by employing previous and future frames of correctly decoded frames, column 7, lines 9-37; the examiner interprets the claimed discontinuity of numbering as the detected error that one of the frame (for example P5 as discussed in column 10, lines 52-63) is unable to be retrieved due to packet loss);

configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the

image frames (the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18); and

use the image frames in the buffer memory in the decoding process (the examiner interprets the process within the dot line as shown in figure 9 as the decoding process, as the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18, therefore, full frame versions of P2, P4, and P6 through P14, as well we P1 and P3 in the buffer memory are use in the decoding process).

(3) Regarding claims 3 and 12:

Apostolopoulos further disclose using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; the examiner interprets “do not exist at a decoder due to the at least one discontinuity in the numbering of the image frame" as due to the error occur in the transmission of the encoded odd frame and prevent the decode block from properly reconstructing P_5, therefore there is a discontinuity in the numbering of image frame, column 9, lines 66-67).

(4) Regarding claim 9:

Apostolopoulos further disclose detecting an error in the compressed video sequence on a basis of a reference in the decoding process to an image frame corresponding to the at least one discontinuity in the numbering of image frames (the error occur in the transmission of the encoded odd frame and prevent the decode block from properly reconstructing P_5, therefore there is a discontinuity in the numbering of image frame, column 9, lines 66-67).

(5) Regarding claims 5 and 14:

Apostolopoulos further disclose continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames (the receiver 332 includes a state recovery 526 for recovering the lost P-frame P5 of the stream 354 using the information provided by the stream 356, the recovery of P5 enables the decode block 322 to continue with its prediction loop by correctly updating its state information and reconstruct p7 to P13, column 10, lines 55-60; as the frames are correctly decode by the decoder, frames are enter into the frame memory as a previous decoded frames as shown in figure 5).

(6) Regarding claims 6 and 15:

Apostolopoulos further discloses entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames (the lost frame may be estimated by replacing it by another correctly

decoded frame, column 11, lines 48-49; the examiner interprets the correctly decode frame that replace the lost frame as the claimed “filler frame”).

(7) Regarding claims 7 and 16:

Apostolopoulos further discloses wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; therefore an odd frame can be replace by an even frame, wherein the even frame number (identifier) indicates that the filler frame do not belong to the odd frame sequence).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) in view of Langberg (US 5,852,630).

(1) Regarding claim 18:

Apostolopoulos disclose a method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding (decode 322 in figure 5 enter the decoded frame into the frame memory including the previous decoded frames as shown in figure 5), the method comprising:

decoding from the compressed video sequence an indication related to at least one discontinuity of numbering of the image frame (the packet is decoded in step 914 and a determination is made whether an error has been detected, when an error is detected, state recovery is performed by employing previous and future frames of correctly decoded frames, column 7, lines 9-37; the examiner interprets the claimed discontinuity of numbering as the detected error that one of the frame (for example P5 as discussed in column 10, lines 52-63) is unable to be retrieved due to packet loss);

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to the art least one discontinuity in the numbering of the image frames (the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18); and

using the image frames in the buffer memory in the decoding process (the examiner interprets the process within the dot line as shown in figure 9 as the decoding process, as the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18, therefore, full frame versions of P2, P4, and P6 through P14, as well we P1 and P3 in the buffer memory are use in the decoding process).

Apostolopoulos discloses all of the subject matter as described above except for the method is perform by a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device.

However, Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Apostolopoulos would have been implemented in a software. The implemented software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg for implementing the method of Apostolopoulos in order to reduce cost and improve the adaptability and flexibility of the communication system.

(2) Regarding claim 20:

Apostolopoulos further disclose using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; the examiner interprets “do not exist at a decoder due to the at least one discontinuity in the numbering of the image frame" as due to the error occur in the transmission of the encoded odd frame and prevent the decode block from properly reconstructing P_5,

therefore there is a discontinuity in the numbering of image frame, column 9, lines 66-67).

(3) Regarding claim 21:

Apostolopoulos disclose all subject matter as discussed in claim 1 but fails to explicitly disclose wherein the number of image frames in the buffer memory are used in sliding widow buffering of image frames.

However, as shown in figure 5, decoded frame as input into the frame memory as previous decode frames, these frames are used for recovery of a lost frame (recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstructed full frame versions of P2, P4, P6 through P14 as well as P1 and P3, column 11, lines 14-18), therefore, the frame memory have to store the decoded frame P2 to P14 in the frame memory on the even frame number sequence and obviously, the frame memory have to store the decoded frame P1 to P13 in the frame memory on the odd frame number sequence, therefore, the examiner interprets the frame memory are used in sliding window buffering of the decode image frame.

It is desirable to have the number of image frames in the buffer memory are used in sliding widow buffering of image frames because the buffer of image frame is need for recovering of a full frame version of a lost frame and provide the benefit of increase the image quality of the recovered video sequence.

(4) Regarding claim 22:

Apostolopoulos further disclose continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames (the receiver 332 includes a state recovery 526 for recovering the lost P-frame P5 of the stream 354 using the information provided by the stream 356, the recovery of P5 enables the decode block 322 to continue with its prediction loop by correctly updating its state information and reconstruct p7 to P13, column 10, lines 55-60; as the frames are correctly decode by the decoder, frames are enter into the frame memory as a previous decoded frames as shown in figure 5).

(5) Regarding claim 23:

Apostolopoulos further discloses entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; the examiner interprets the correctly decode frame that replace the lost frame as the claimed “filler frame”).

(6) Regarding claim 24:

Apostolopoulos further discloses wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; therefore an odd frame can be replace by an even

frame, wherein the even frame number (identifier) indicates that the filler frame do not belong to the odd frame sequence).

(7) Regarding claim 25:

Apostolopoulos further discloses wherein configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by frame number, whereby no data is entered into the buffer memory (past and future frames with respect to the lost frame P5 to recover frame P5, column 10, lines 60-63, as the past and future frames are identified by frame number and when the decoder cannot decode P5, the decode block 322 cannot reconstruct P5 to P13, therefore, no data is entered into the buffer memory (frame memory that store previous decoded frames)).

Apostolopoulos fails to disclose using memory indications for reconstruction of a lost frame.

However, as shown in figure 5, the correctly decoded frames are storage in the frame memory as previous decoded frames. Therefore, it would have been obvious to one of ordinary skill in the art to modify the invention Apostolopoulos to use memory indication (location) for locating future frames in the even frame number sequence and past frame in the odd frame number sequence for reconstruction of the lose frame since using the frame number and memory location are interchangeable to one of ordinary skill in the art.

7. Claims 4, 8, 13, and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2).

(1) Regarding claim 4 and13:

Apostolopoulos disclose all subject matter as discussed in claim 1 but fails to explicitly disclose wherein the number of image frames in the buffer memory are used in sliding widow buffering of image frames.

However, as shown in figure 5, decoded frame as input into the frame memory as previous decode frames, these frames are used for recovery of a lost frame (recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstructed full frame versions of P2, P4, P6 through P14 as well as P1 and P3, column 11, lines 14-18), therefore, the frame memory have to store the decoded frame P2 to P14 in the frame memory on the even frame number sequence and obviously, the frame memory have to store the decoded frame P1 to P13 in the frame memory on the odd frame number sequence, therefore, the examiner interprets the frame memory are used in sliding window buffering of the decode image frame.

It is desirable to have the number of image frames in the buffer memory are used in sliding widow buffering of image frames because the buffer of image frame is need for recovering of a full frame version of a lost frame and provide the benefit of increase the image quality of the recovered video sequence.

(2) Regarding claims 8 and 17:

Apostolopoulos further discloses wherein configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by frame number, whereby no data is entered into the buffer memory (past and future frames with respect to the lost frame P5 to recover frame P5, column 10, lines 60-63, as the past and future frames are identified by frame number and when the decoder cannot decode P5, the decode block 322 cannot reconstruct P5 to P13, therefore, no data is entered into the buffer memory (frame memory that store previous decoded frames)).

Apostolopoulos fails to disclose using memory indications for reconstruction of a lost frame.

However, as shown in figure 5, the correctly decoded frames are storage in the frame memory as previous decoded frames. Therefore, it would have been obvious to one of ordinary skill in the art to modify the invention Apostolopoulos to use memory indication (location) for locating future frames in the even frame number sequence and past frame in the odd frame number sequence for reconstruction of the lose frame since using the frame number and memory location are interchangeable to one of ordinary skill in the art.

Allowable Subject Matter

8. Claims 2, 11, and 19 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU M. LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 7:30-4:00 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Examiner, Art Unit 2611
2/14/2011

Notice of References Cited	Application/Control No. 11/338,996	Applicant(s)/Patent Under Reexamination HANNUKSELA, MISKA	
	Examiner SIU M. LEE	Art Unit 2611	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-7,103,669	09-2006	Apostolopoulos, John G.	709/231
*	B	US-5,852,630	12-1998	Langberg et al.	375/219
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

SEARCHED

Class	Subclass	Date	Examiner
375	240.27, 240.25	6/14/2010	Siu M. Lee

SEARCH NOTES

Search Notes	Date	Examiner
EAST	6/14/2010	Siu M. Lee
Discuss the application with Chieh Fan	6/14/2010	Siu M. Lee
EAST	11/1/2010	Siu M. Lee
Consult John Peng on the 101 problem in claim 10-17	11/1/2010	Siu M. Lee
Discuss the amendment with Cheih Fan	10/27/2010	Siu M. Lee
EAST	2/14/2011	Siu M. Lee

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	06/13/2010	11/01/2010	02/14/2011						
	1	✓	✓	✓						
	2	✓	✓	O						
	3	✓	✓	✓						
	4	✓	✓	✓						
	5	✓	✓	✓						
	6	✓	✓	✓						
	7	O	O	✓						
	8	O	O	✓						
	9	O	✓	✓						
	10	✓	✓	✓						
	11	✓	✓	O						
	12	✓	✓	✓						
	13	✓	✓	✓						
	14	✓	✓	✓						
	15	✓	✓	✓						
	16	✓	O	✓						
	17	✓	O	✓						
	18	✓	✓	✓						
	19	✓	✓	O						
	20	✓	✓	✓						
	21	✓	✓	✓						
	22	✓	✓	✓						
	23	✓	✓	✓						
	24	✓	O	✓						
	25	✓	O	✓						
	26	✓	-	-						
	27	✓	-	-						
	28	✓	-	-						
	29	✓	-	-						
	30	✓	-	-						
	31	✓	-	-						

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 5361

SERIAL NUMBER	FILING or 371(c) DATE RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/338,996	01/25/2006	375	2611	TPO.004.C1

APPLICANTS
Miska Hannuksela, Tampere, FINLAND;

**** CONTINUING DATA ***************************
This application is a CON of 10/306,942 11/29/2002 PAT 7,894,521

**** FOREIGN APPLICATIONS ***************************
FINLAND 20020127 01/23/2002

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****
02/28/2006

Foreign Priority claimed ☑ Yes ☐ No 35 USC 119(a-d) conditions met ☑ Yes ☐ No Verified and Acknowledged /SIU M LEE/ Examiner's Signature	☐ Met after Allowance Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		FINLAND	4	31	6

ADDRESS
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

TITLE
Grouping of image frames in video coding

FILING FEE RECEIVED	FEES: Authority has been given in Paper	☐ All Fees
2150	No._____ to charge/credit DEPOSIT ACCOUNT	☐ 1.16 Fees (Filing)
	No._____ for following:	☐ 1.17 Fees (Processing Ext. of time)
		☐ 1.18 Fees (Issue)
		☐ Other _____
		☐ Credit

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L3	22	buffer$3 with slid$3 window with video	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 09:32
L4	6	3 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 09:32
L5	1	"7103669".pn. and replac$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 09:50
L6	1	11/338996 and intentional	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 09:55
L7	682	us$3 with memory (location or indicat$3) with frame	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:18
L8	2	7 and frame reconstruction	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:18
L9	64	memory (location or indicat$3) with frame number	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:23

L10	57	9 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:23
L11	0	10 and (memory (location or indicat$3) with frame number with instead)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:23
L12	0	10 and (memory (location or indicat$3) with frame number with reaplc$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:23
L13	1	10 and (memory (location or indicat$3) with frame number with replac$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:23
S1	0	"7103669".pn. and (memory or buffer)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 17:15
S2	1	"7103669".pn. and (stor$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 17:15
S3	11	sliding window buffering	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 17:32
S4	6	S3 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 17:34
S5	0	"7103669".pn. and (frame near2 memory)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 19:35

S6	1	"7103669".pn. and (previous frame)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 19:35

EAST Search History (Interference)

< This search history is empty>

2/ 14/ 2011 10:52:52 AM
C:\ Documents and Settings\ slee7\ My Documents\ EAST\ Workspaces\ 11338996 201102.wsp

To: tdotter@hfiplaw.com,roswood@hfiplaw.com,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 10888

Feb 18, 2011 06:01:49 AM

Dear PAIR Customer:

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 10888 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
11338996	CTNF	02/18/2011	TPO.004.C1
	892	02/18/2011	TPO.004.C1

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
11/338,996		2611	

OC000000048476017

Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 10888 on 11/30/2010

- **Correspondence Address**
- **Power of Attorney Address**

The address of record for Customer Number 10888 is:

10888
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	2002	Hannuksela, "Signaling of Enhanced GOPs", Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 4th Meeting, Austria, July 22-26, 2002.
	4/5/2011	File History for EP Application No. 03700321.7 as retrieved from the European Patent Office Electronic File System on 4/5/2011, 222 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

vom Absender auszufüllen / A remplir par l'expéditeur / Completed by the sender

Sendungsnummer / No de l'envoi / Item number	Art der Sendung / Nature de l'envoi / Type	
	X	Brief / Lettre / Letter
		Paket / Colis / Parcel
	X	Einschreiben / Recommandé / Registered Mail
		Wertangabe / Valeur déclarée / Value :
		Gewöhnliche Anweisung / Mandat ordinaire / Advice of inscription :
		Zahlkarte / de versement / Advice of payment
		Auslieferungsnachweis / Chèque d'assignation / Proof of delivery

Datum: _____

27. 01. 11

Empf Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy
Hermiankatu 1 B
33720 Tampere
FINLANDE

oi / Article adressed to

RK 19 214 747 5DE

Page 1855 of 2504

Am Bestimmungsort auszufüllen / A compléter à destination / Completed at destination

O.g. Sendung/Betrag wurde ordnungsgemäß / L'envoi mentionné ci-dessus a été dûment / The article mentioned above was duly

☐ ausgehändigt / remis / delivered ☐ ausgezahlt / payé / paid ☐ dem Postbankkonto gutgeschrieben / inscrit en CCP / credited to bank account

* Diese(r) Schein/Anzeige kann vom Empfänger oder wenn die Vorschriften des Bestimmungslandes dies vorsehen, von einem Beauftragten oder vom Mitarbeiter des Postunternehmens im Bestimmungsland unterschrieben werden.
* Cet avis pourra être signé par le destinataire ou, si les règlements du pays de destination le prévoient, par une autre personne autorisée au par l'agent du bureau de destination.
* This receipt must be signed by the adressee or a person authorized to sign under the regulations of the country of destination or if those regulations so provide, by the employee of the office of destination.

1.2.2011 KARLSSON

Datum und Unterschrift* / Date et signature* / Date and signature*

Deutsche Post AG
Entreprise des Postes allemandes S.A.

Postsache
Service des Postes

Anmeldung Nr./Application No./Demande n°.
03700321.7 - 2223 / 1479245

Rückschein / Auszahlungsschein / Gutschriftsanzeige
Avis de réception / Avis de paiement / Avis d'inscription
Advice of delivery / Advice of payment / Credit advice

CN07

AR

TAMBERET 27
-1-211-9

Stempel des Postamts, das den Schein zurücksendet.

Timbre du bureau renvoyant l'avis

Postmark of the office returning the receipt

Luftpost
Prioritaire Par avion

Zurücksenden an / Renvoyer à / Return to

Europäisches Patentamt

80298 MÜNCHEN

Deutschland / l'Allemagne / Germany

915-005-000 05.05

Europäisches Patentamt
European Patent Office
Office européen des brevets

European Patent Office
80298 MUNICH
GERMANY
Tel. +49 (0)89 2399 - 0
Fax +49 (0)89 2399 - 4465

Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy
Hermiankatu 1 B
33720 Tampere
FINLANDE

Formalities Officer
Name: Exner, Karin
Tel.: 7826
or call:
+31 (0)70 340 45 00

Date: 27-01-2011

Reference	Application No./Patent No.
TP103660	03700321.7 - 2223 / 1479245

Applicant/Proprietor: Nokia Corporation

EPA/EPO/OEB Formblatt/Form/Formulaire : 2008/ 2906

Empfangsbescheinigung über den Zugang des vorstehend bezeichneten Schriftstücks
Acknowledgement of receipt of the document specified above
Récépissé du document spécifié ci-dessus

Unter Bezugnahme auf die Mitteilung im ABl EPA 7/2010, 377 wird gebeten, die Empfangsbescheinigung mit Empfangsdatum und Unterschrift zu versehen und **umgehend** an das EPA zurückzusenden:

With reference to the Notice in OJ EPO 7/2010, 377, you are requested to date and sign the acknowledgement of receipt and return it to the EPO **immediately**:

Conformément au communiqué paru au JO OEB 7/2010, 377, vous êtes prié d'indiquer sur le récépissé la date de réception du document, de signer le récépissé et de le renvoyer **sans délai** à l' OEB:

- **über die Online-Dienste des EPA** (als Anlage zu EPA Form 1038) / **through EPO Online Services** (as annex to EPO Form 1038) / **par les services en ligne de l'OEB** (en tant que pièce jointe au formulaire OEB 1038),
- **per Fax / by fax / par téléfax (+49 (0) 89 2399-4465 or +31 (0) 70 340-3016)**
- oder per Post / or by post / ou par courrier.

Empfangen am / Received on / Reçu le :

Unterschrift / Signature:

Empfangsberechtigter/authorised recipient/
le destinataire ou la personne dûment mandatée

Rücksende-Adresse / Return address / Adresse de retour
(Umschlag / envelope / enveloppe ISO C4 / DL / C6/C5 / C6)

Europäisches Patentamt
80298 MÜNCHEN
DEUTSCHLAND

Rücksende-Adresse / Return address / Adresse de retour

EPA/EPO/OEB Form 2936 08.10

KE23274

The examination is being carried out on the **following application documents**

Description, Pages

All requests

1-34	as published

Drawings, Sheets

All requests

1/4-4/4	as published

Claims, Numbers

Main Request

1-25	filed in electronic form on	13-10-2010

Auxiliary Request 1

1-21	filed in electronic form on	13-10-2010

Auxiliary Request 2

1-17	filed in electronic form on	13-10-2010

1 The applicant is summoned to attend oral proceedings before the Examining Division in accordance with Article 116(1) and Rule 115 EPC. The points to be discussed are the following:

2 *Added subject-matter*

2.1 *Main request*

Claim 1 has been amended to include *inter alia* the following features:

a. the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other; and

b. the information indicative of the sub-sequence is independent of the frame format.

a. Firstly, it is not clear what it means that the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are *different* from each other. If the two pieces of information indicate different things, they are inherently different. Secondly, there is no direct and unambiguous basis in the application as filed for this language. On the contrary, in paragraph [0033], it appears that the different sub-sequences and their associated information depend on each other.

b. It is unclear what is meant by the information indicative of the sub-sequence being *independent* of the frame format, especially since the formation of the sub-sequences depends on the frame format. Furthermore, there is no direct and unambiguous basis in the application as filed for this feature.

As basis for the amendment, the applicant refers to paragraphs [0032] *"the sub-sequences of each scalability layer are assigned a unique identifier"* and [0031] *"each scalability layer is assigned a unique identifier such as a layer number"*. The Examining Division does not consider this to be a clear and unequivocal basis for the present wording of claim 1.

For these reasons, claim 1 contravenes article 123(2) EPC and is, therefore, not allowable. The same applies for claims 11, 16, 20, 24 and 25 which are not allowable either.

2.2 *1st auxiliary request*

Claims 1, 10, 14, 17, 20 and 21 of the 1st auxiliary request contravene, for the same reasons, Article 123(2) EPC and are, therefore, not allowable.

2.3 *2nd auxiliary request*

Claims 1, 9, 12, 16 and 17 of the 2nd auxiliary request contravene, for the same reasons, Article 123(2) EPC and are, therefore, not allowable.

3 *Priority*

The present application claims priority from FI20020127 (P1) filed on 23.01.2002 and from US10/206942 (P2) filed on 29.11.2002.

The feature of including the information indicative of the sub-sequence as Supplemental Enhancement Information is not disclosed in P1. The effective date of this subject-matter is, therefore, the filing date of P2, i.e. **29.11.2002** (Article 87(1) and 89 EPC).

4 *Prior art*

As a consequence, the following documents are prior art under Article 54(2) EPC for the following claims:

D2 HANNUKSELA: "Enhanced Concept of GOP", ITU STUDY GROUP 16 - VIDEO CODING EXPERTS GROUP - ISO/IEC MPEG & ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 AND ITU-T SG16 Q6), XX, XX, no. JVT-B042r1, 1 February 2002 (2002-02-01), XP030005042,

D3 HANNUKSELA: "Signaling of Enhanced GOPs", ITU STUDY GROUP 16 - VIDEO CODING EXPERTS GROUP - ISO/IEC MPEG & ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 AND ITU-T SG16 Q6), XX, XX, no. JVT-D098, 26 July 2002 (2002-07-26), XP030005372,

Main request: claims 4, 14, 18, 22

1st auxiliary request: claims 1, 10, 14, 17, 20, 21

2nd auxiliary request: 1, 9, 12, 14, 16, 17

Copies of the documents are annexed to this communication.

5 *Novelty*

D3 is a document by the present inventor and discloses the organization of pictures in layers and sub-sequences (page 2, s. 3.1), wherein a sub-sequence represents a number of inter-dependent pictures that can be disposed without affecting the decoding of any other sub-sequence in less enhanced layers (page 1, s. 1). The information indicative of the layers and sub-sequences is signalled in the Supplemental Enhancement Information (SEI) message (page 3, s. 3.4).

The subject-matter of the following claims, therefore, lack novelty over D3 in the sense of Article 54(1) and (2) EPC, contrary to Article 52(1) EPC.

Main request: claims 4, 14, 18, 22

1st auxiliary request: claims 1, 10, 14, 17, 20, 21

2nd auxiliary request: 1, 9, 12, 14, 16, 17

6 *Inventive step*

If the applicant files amended claims which overcome the objections in point 2 *supra*, novelty and inventive step (Article 52(1), 54 and 56 EPC) will be discussed with respect to D1 and, where applicable, D2 and D3. The applicant should be prepared to indicate the basis in the priority document P1 for the subject-matter of the amended claims.

D1 discloses temporally scalable video coding (fig. 2; page 3, lines 15-27). The base layer (fig. 2, e.g. base 10fps) forms a first sub-sequence comprising I and P pictures and the enhancement layer (e.g. Enh 30fps) one or several further sub-sequences comprising P-pictures. Information indicative of the picture dependencies and, consequently of the respective sub-sequences, is signalled in the picture header (page 4, lines 2-6). Furthermore, as an alternative, D1 suggests to include each temporal layer (sub-sequence) in a separate transport stream (page 5, last paragraph). Information indicative of the respective sub-sequences would therefore be present at the transport layer.

It appears that any difference with respect to D1 would lie in the particular syntax for indicating the respective sub-sequences. This is in the Examining Division's opinion merely a matter of design. In principle, a coding choice or concept can be indicated at any appropriate level of the bitstream. The person skilled in the art is well aware of the trade-off between flexibility and the amount of overhead and will choose the appropriate syntax depending on the circumstances, including the intended application and compatibility issues with various standards. Therefore, it is not at present apparent which part of the application could serve as a basis for a new, allowable claim.

7 *Clarity*

Claims 1, 9, 12, 14, and 16 of the second auxiliary request include the feature that the information indicative of the sub-sequence comprises a *"scalability layer identifier"* and a *"sub-sequence identifier"*. The scalability layers are not defined in said claims and it is unclear how they differ from the sub-sequences (see point 6 above with reference to D1: a sub-sequence can be a temporal scalability layer; see also the description, paragraphs [0017] and [0032]: the second sub-sequence is removable without affecting decodability of the

remaining video sequence, which is, in fact, the definition of a scalability layer). Claims 1, 9, 12, 14 and 16 of the second auxiliary request, therefore, lack clarity, contrary to Article 84 EPC.

Europäisches Patentamt
European Patent Office
Office européen des brevets

European Patent Office
80298 MUNICH
GERMANY
Tel: +49 89 2399 0
Fax: +49 89 2399 4465

Application No.: 03 700 321.7

Preparation for oral proceedings - Instructions to Support Service

Oral proceedings are to be held in connection with the above patent application

1. The matters to be discussed are set out in the annex (Form 2906)

2. Dispatch the summons using Form 2008/2310 and Form 2906 for the parties to attend on:

Day 26.05.2011 Time 09:00

ROOMS

Room booked

ORAL 01, 02, 03 and 05 coded

10.09.11 EK
Date Initials

2.1 Parties' submissions in preparation for the oral proceedings, if any, should be made no later than

1 month(s)

before the date of the oral proceedings
(transfer to Form 2008.1 / 2310.1)

2.2 Encode ORAL(04)

coded 10.09.11 EK
Date Initials

2.3 Dispatch Form 2008.7 / 2310.7 to division

10.09.11 EK
Date Initials

3. ☐ Arrange for the following special equipment to be provided in the conference room:

Date Initials

Europäisches Patentamt
European Patent Office
Office européen des brevets

European Patent Office
80298 MUNICH
GERMANY
Tel: +49 89 2399 0
Fax: +49 89 2399 4465

4. Request language service to provide simultaneous interpretation facilities as necessary

..... Date ..... Initials

5. Return the dossier to primary examiner with Form 2041 (15 days before the oral proceedings)

..... Date ..... Initials

6. Check that summons has been received (Form 2936 / advice of delivery)

7. 15 days before the oral proceedings:
- dispatch the dossier to the primary examiner.

21.12.2010
Date

Willems, Brigitte	Hindelang, Thomas	Wahrenberg, Annika	
Chairman	2nd examiner	1st examiner	Legal member

Enclosure(s):
XP30005042, XP30005372

Europäisches Patentamt
European Patent Office
Office européen des brevets

European Patent Office
80298 MUNICH
GERMANY
Tel. +49 (0)89 2399 - 0
Fax +49 (0)89 2399 - 4465

Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy
Hermiankatu 1 B
33720 Tampere
FINLANDE

Formalities Officer
Name: Simão Exner, Karin
Tel.: 8637 -7826
or call:
+31 (0)70 340 45 00

Date
27-01-2011

Reference	Application No./Patent No.
TP103660	03700321.7 - 2223 / 1479245
Applicant/Proprietor	
Nokia Corporation	

Summons to attend oral proceedings pursuant to Rule 115(1) EPC

You are hereby summoned to attend oral proceedings arranged in connection with the above-mentioned European patent application.

The matters to be discussed are set out in the communication accompanying this summons (EPO Form 2906).

The oral proceedings, which will not be public, will take place before the Examining Division.

on 26.05.11 at 09.00 hrs at the EPO,
PschorrHöfe, Bayerstr. 34, D-80335 München

No changes to the date of the oral proceedings can be made, except on serious grounds (see OJ EPO 1/2009, 68). If you do not appear as summoned, the oral proceedings may continue without you (R. 115(2) EPC, see also OJ EPO 10/2008, 471).

Your attention is drawn to Rule 4 EPC, regarding the language of the oral proceedings, and to the Special edition No. 3 OJ EPO 2007, L.1., concerning the filing of authorisations for company employees and lawyers acting as representatives before the EPO.

The final date for making written submissions and/or amendments (R. 116 EPC) is 26.04.11.

The actual room number as well as the waiting room numbers will be given to you by the porter in the foyer at the above EPO address.

Parking is available free of charge in the underground car park. However, this applies only in the case of accessing the car park via the entrance "Zollstrasse".

1st Examiner:	2nd Examiner:	Chairman:
Wahrenberg I	Hindelang T	Willems B

For the Examining Division

Annexes:
Confirmation of receipt (Form 2936)
Communication (EPO Form 2906)

Europäisches Patentamt • European Patent Office • Office européen des brevets

Registered letter with advice of delivery

to EPO postal service: 19.01.11

Europäisches Patentamt
European Patent Office
Office européen des brevets

European Patent Office
80298 MUNICH
GERMANY
Tel. +49 (0)89 2399 - 0
Fax +49 (0)89 2399 - 4465

Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy
Hermiankatu 1 B
33720 Tampere
FINLANDE

Formalities Officer
Name: Simão
Tel.: 8627 8242
or call:
+31 (0)70 340 45 00

Schalinatus, Dana

in advance by fax: 358 / 10 227 2662

Date
05-11-2010

Reference	Application No./Patent No.
TP103660	03700321.7 - 2223 / 1479245
Applicant/Proprietor	
Nokia Corporation	

BRIEF COMMUNICATION

Oral Proceedings on 17.11.10

Subject: ☒ Your letter of 13.10.2010

Communication:
- ☒ The summons to attend oral proceedings on 17.11.10 has been cancelled.
- ☒ The procedure will be continued in writing.
- ☐ The date fixed for oral proceedings is maintained.
- ☐ A new date will be set later.
- ☐

Please take note.

For the Examining Division

Europäisches Patentamt • European Patent Office • Office européen des brevets

European patent application No. 03700321.7

II. To the Formalities Officer

1. ☐ The date / time fixed for oral proceedings is maintained.

   ☐ (if necessary) The reasons are indicated on enclosed EPO Form 2906/2906O, which is to be dispatched to the party with EPO Form 2008A/2310A.

2. ☐ At the instigation of the division the oral proceedings cannot take place on the arranged date for the reasons indicated on enclosed EPO Form 2906, which is to be dispatched to the party with EPO Form 2008A/2310A.

   Date _____ Director (see Internal Instructions IE-III, 1.2) _____

3. ☒ The summons to attend oral proceedings on 17.11.10 should be cancelled.

   3.1 ☒ The proceedings will be continued in writing.

   3.2 A new date will be set later.

   3.3 ☐ A new date/time is set as follows:

   date ..... at ..... hrs.
   Parties' written submissions and amendments in preparation for the oral proceedings, if any, should be made not later than

   ..... month(s) before the date of the oral proceedings.

4. ☐ The request for the oral proceedings to be held as a videoconference

   4.1 ☐ is allowed.

   4.2 ☐ is rejected for the reasons indicated on enclosed EPO Form 2906, which is to be dispatched to the party with EPO Form 2008A.

5. ☐ The application is deemed to be withdrawn. The oral proceedings are cancelled.

29.10.10
Date

Chairman _____ Second member _____ Primary Examiner _____ Legal member _____

III. Action taken by the formalities officer

☒ EPO Form 2008A/2310A has been dispatched, if applicable (see point II above) together with EPO Form 2906/2906O.

☐ A new summons has been dispatched.

☐ .....

02.11.10 Sd
Date

D. Schalnatus
Formalities Officer

EPO Form 2088 04.10

IS23754

Europäisches Patentamt
European Patent Office
Office européen des brevets

European patent application No. 03700321.7

URGENT !
To be completed (under II.) and returned to the formalities officer immediately

Maintenance / Change of date / Cancellation of oral proceedings arranged for: day 17.11.10 at 09.30 hrs

I. To the Examining / Opposition division

1. ☐ The proprietor has requested the revocation (or the like) of his patent (date: .....).

2. ☑ The party / witness .representative..... has indicated (date: 14.10.2010.....) that he / she

   ☐ requests the date / time of oral proceedings to be changed.

   ☑ withdraws the request for oral proceedings. NO !

   ☐ will not be attending.

   ☐ wishes to be heard by a national court.

   ☑ has filed amendments (in examination proceedings).

   ☐ has requested that the oral proceedings be held as a videoconference (in examination proceedings).

3. ☐ Two months before the oral proceedings the application is deemed to be withdrawn (in examination proceedings). The oral proceedings have to be cancelled.

15-10-2010
Date

Simão, Iolanda
Formalities Officer

EPO Form 2088 04.10

IS23754

1ST AUXILIARY REQUEST:

CLAIMS

1. A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising the steps of

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

2. A method according to claim 1, **characterized** by

forming into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

3. A method according to claim 1 or 2, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

4. A method according to any preceding claim, **characterized** in that

the second sub-sequence being removable from the video sequence without affecting decodability of the remaining video sequence.

25. A method according to any preceding claim ~~1~~, **characterized** by comprising the steps of

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

~~3~~6. A method according to claim 25, **characterized** by comprising the steps of

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from ~~a group comprising~~

a video frame of ~~a higher~~the scalability layer containing the first sub-sequence, and/or

a video frame of another sub-sequence in the same scalability layer.

~~4~~7. A method according to claim ~~3~~6, **characterized** by comprising the step of

determining the dependencies of the video frames of the second sub-sequence on the basis of at least ~~a~~the scalability layer identifier and ~~a~~the sub-sequence identifier.

~~5~~8. A method according to claim ~~3~~6 or 47, **characterized** by comprising the step of

coding ~~the~~a first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

~~6~~9. A method according to claim ~~3~~6 or 47, **characterized** by comprising the step of

coding ~~the~~a first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

~~7. A method according to claim 3, characterized by comprising the step of~~

determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.

8. A method according to claim 3, **characterized** by comprising the step of

determining a unique identifier for each video frame on the basis of the image number.

9. A method according to claim 7 or 8, **characterized** by comprising the step of

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. A method according to any one of the preceding claims, **characterized** in that

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

11. A method according to any one of the preceding claims, **characterized** by comprising the step of

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

12. A method according to any one of the preceding claims, **characterized** by comprising the step of

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

13. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to control the bit rate of a video sequence transmission in which a first sub-sequence of the video sequence comprises video frames of at least a first frame format and at least a second sub-sequence comprises video frames of at least a second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

~~the bit rate control being arranged to be carried out in at least one of the following ways:~~

~~at least one sub-sequence is removed from the video sequence,~~

~~at least one sub-sequence is added to the video sequence.~~

~~11~~10. A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; ~~and to~~

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

include the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format. ~~and to~~

11. The video encoder according to claim 10, wherein the video encoder is arranged to

form into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

12. The video encoder according to claim 10 or 11, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

13. The video encoder according to any of the claims 10 - 12, **characterized** in that the video encoder is arranged to

encode the second sub-sequence to be removable from the video sequence without affecting decodability of the remaining video sequence.

14. A method for controlling bit rate of a video sequence transmission in a network element of a telecommunications system, **characterized** by

receiving, in the network element, a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

obtaining information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of the received video sequence;

controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

15. The method according to claim 14, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

16. A method according to claim 14 or 15, **characterized** by comprising

deducing on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

removing the second sub-sequence.

17. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to receive a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

the element is arranged to obtain information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of the received video sequence; and

the element is arranged to control the bit rate of at least one transmission of said video sequence on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

at least one sub-sequence is removed from the video sequence,

at least one sub-sequence is added to the video sequence.

18. The element according to claim 17, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

19. The element according to claim 17 or 18, **characterized** in that the element is arranged to

deduce on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

remove the second sub-sequence.

~~15. A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising the steps of~~

~~decoding a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;~~

~~decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;~~

~~determining dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and~~

~~reconstructing at least part of the video sequence on the basis of the dependencies of the sub-sequences.~~

~~16. A method according to claim 15, wherein video frames are entered into a sliding buffer memory in connection with decoding, **characterized** by comprising the steps of~~

~~decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;~~

~~configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and~~

~~continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame~~

number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

17. A method according to claim 16, **characterized** by comprising the step of

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

18. A method according to any one of claims 15 to 17, **characterized** by comprising the step of

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

19. A method according to any one claims 15 to 18, **characterized** by comprising the steps of

initiating the decoding from a random point in the video sequence;

determining an independently decodable sub-sequence that is next after the random point, and

setting the value of the image number of the first video frame in the sub-sequence at zero.

20. A method according to any one claims 15 to 19, **characterized** by comprising the steps of

identifying at least partly temporally overlapping sub-sequence frames from the video sequence based on the picture order count information, and

outputting from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.

21. A video decoder for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video decoder is arranged to

decode a first sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the first frame format;

decode at least a second sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the second frame format,

~~and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;~~

~~determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and~~

~~reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.~~

~~22~~20. A computer program executable in a data processor for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **c h a r a c t e r i z e d** in that the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; ~~and~~

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format. ~~and~~

~~a program code for determining data into the video sequence to determine which video frames belong to the second sub-sequence.~~

21. A computer program executable in a data processor for controlling bit rate of a video sequence transmission, **characterized** in that the computer program comprises

a program code for obtaining information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of a received video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format; and

a program code for controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

~~23. A computer program for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the computer program comprises~~

~~a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and~~

a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.

24. A video signal comprising a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, characterized in that the video signal comprises

a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format;

at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

at least one data field determining the video frames that belong to the second sub-sequence.

2ND AUXILIARY REQUEST:

CLAIMS

1. A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising the steps of

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier, the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

2. A method according to claim 1, **characterized** by

forming into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

3. A method according to claim 1 or 2, wherein

the second sub-sequence being removable from the video sequence without affecting decodability of the remaining video sequence.

24. A method according to any preceding claim 1, **characterized** by comprising the steps of

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

35. A method according to claim 24, **characterized** by comprising the steps of

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from ~~a group comprising~~

a video frame of ~~a higher~~the scalability layer containing the first sub-sequence, and/or

a video frame of another sub-sequence in the same scalability layer.

46. A method according to claim 35, **characterized** by comprising the step of

determining the dependencies of the video frames of the second sub-sequence on the basis of at least ~~a~~ the scalability layer identifier and ~~a~~ the sub-sequence identifier.

57. A method according to claim ~~3~~ 5 or 46, **characterized** by comprising the step of

coding ~~the~~ a first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

68. A method according to claim ~~3~~ 5 or 46, **characterized** by comprising the step of

coding ~~the~~ a first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

~~7. A method according to claim 3, characterized by comprising the step of~~

~~determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.~~

~~8. A method according to claim 3, characterized by comprising the step of~~

determining a unique identifier for each video frame on the basis of the image number.

9. A method according to claim 7 or 8, **characterized** by comprising the step of

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. A method according to any one of the preceding claims, **characterized** in that

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

11. A method according to any one of the preceding claims, **characterized** by comprising the step of

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

12. A method according to any one of the preceding claims, **characterized** by comprising the step of

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

13. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to control the bit rate of a video sequence transmission in which a first sub-sequence of the video sequence comprises video frames of at least a first frame format and at least a second sub-sequence comprises video frames of at least a second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

the bit rate control being arranged to be carried out in at least one of the following ways:

at least one sub-sequence is removed from the video sequence,

at least one sub-sequence is added to the video sequence.

~~149~~. A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **c h a r a c t e r i z e d** in that the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; ~~and to~~

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

include the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier, the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format. ~~and to~~

10. The video encoder according to claim 9, wherein the video encoder is arranged to

form into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

11. The video encoder according to claim 10 or 11, wherein the video encoder is arranged to

encode the second sub-sequence to be removable from the video sequence without affecting decodability of the remaining video sequence.

12. A method for controlling bit rate of a video sequence transmission in a network element of a telecommunications system, **characterized** by

receiving, in the network element, a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

obtaining information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of the received video sequence, the information indicative of the sub-sequence comprising a scalability layer identifier and a sub-sequence identifier;

controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

13. The method according to claim 12, **characterized** by comprising

deducing on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

removing the second sub-sequence.

14. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to transmit compressed video sequences to at least one terminal device and to

control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to receive a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

the element is arranged to obtain information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of the received video sequence, the information indicative of the sub-sequence comprising a scalability layer identifier and a sub-sequence identifier; and

the element is arranged to control the bit rate of at least one transmission of said video sequence on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

at least one sub-sequence is removed from the video sequence,

at least one sub-sequence is added to the video sequence.

15. The element according to claim 14, **characterized** in that the element is arranged to

deduce on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

remove the second sub-sequence.

~~15. A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from~~

at least one of the other video frames, characterized by comprising the steps of

decoding a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;

decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determining dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and

reconstructing at least part of the video sequence on the basis of the dependencies of the sub-sequences.

16. A method according to claim 15, wherein video frames are entered into a sliding buffer memory in connection with decoding, characterized by comprising the steps of

decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;

configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

17. A method according to claim 16, characterized by comprising the step of

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

18. A method according to any one of claims 15 to 17, characterized by comprising the step of

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

~~19. A method according to any one claims 15 to 18, characterized by comprising the steps of~~

~~initiating the decoding from a random point in the video sequence;~~

~~determining an independently decodable sub-sequence that is next after the random point, and~~

~~setting the value of the image number of the first video frame in the sub-sequence at zero.~~

~~20. A method according to any one claims 15 to 19, characterized by comprising the steps of~~

~~identifying at least partly temporally overlapping sub-sequence frames from the video sequence based on the picture order count information, and~~

~~outputting from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.~~

~~21. A video decoder for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, characterized in that the video decoder is arranged to~~

~~decode a first sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the first frame format;~~

~~decode at least a second sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;~~

~~determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and~~

~~reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.~~

~~22~~16. A computer program executable in a data processor for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being

predicted from at least one of the other video frames, **c h a r a c t e r i z e d** in that the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; ~~and~~

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier, the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format. ~~and~~

~~a program code for determining data into the video sequence to determine which video frames belong to the second sub-sequence.~~

17. A computer program executable in a data processor for controlling bit rate of a video sequence transmission, **c h a r a c t e r i z e d** in that the computer program comprises

a program code for obtaining information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of a received video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of

which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier; and

a program code for controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

~~23. A computer program for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, characterized in that the computer program comprises~~

~~a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and~~

~~a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and~~

~~a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and~~

~~a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.~~

24. A video signal comprising a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, characterized in that the video signal comprises

a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format;

at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

at least one data field determining the video frames that belong to the second sub-sequence.

MAIN REQUEST:

CLAIMS

1. A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising the steps of

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; ~~and~~

wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

2. A method according to claim 1, **characterized** by

forming into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

3. A method according to claim 1 or 2, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

4. A method according to any preceding claim, **characterized** by comprising the step of

including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used

for transferring the video sequence, or as Supplemental Enhancement Information (SEI).

5. A method according to any preceding claim, **characterized** in that

the second sub-sequence being removable from the video sequence without affecting decodability of the remaining video sequence.

~~2~~6. A method according to any preceding claim ~~1~~, **characterized** by comprising the steps of

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

~~3~~7. A method according to claim ~~2~~6, **characterized** by comprising the steps of

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from ~~a group comprising~~

a video frame of ~~a higher~~the scalability layer containing the first sub-sequence, and/or

a video frame of another sub-sequence in the same scalability layer.

~~4~~8. A method according to claim ~~3~~7, **characterized** by comprising the step of

determining the dependencies of the video frames of the second sub-sequence on the basis of at least ~~a~~the scalability layer identifier and ~~a~~the sub-sequence identifier.

~~5~~9. A method according to claim ~~3~~7 or ~~4~~8, **characterized** by comprising the step of

coding ~~the~~a first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

~~6~~10. A method according to claim ~~3~~7 or ~~4~~8, **characterized** by comprising the step of

coding ~~the~~a first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

7. A method according to claim 3, **characterized** by comprising the step of

determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.

8. A method according to claim 3, **characterized** by comprising the step of

determining a unique identifier for each video frame on the basis of the image number.

9. A method according to claim 7 or 8, **characterized** by comprising the step of

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. A method according to any one of the preceding claims, **characterized** in that

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

11. A method according to any one of the preceding claims, **characterized** by comprising the step of

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

12. A method according to any one of the preceding claims, **characterized** by comprising the step of

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

13. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to control the bit rate of a video sequence transmission in which a first sub-sequence of the video sequence comprises video frames of at least a first frame format and at least a second sub-sequence comprises video frames of at least a second frame format, at least

~~one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;~~

~~the bit rate control being arranged to be carried out in at least one of the following ways:~~

~~at least one sub-sequence is removed from the video sequence,~~

~~at least one sub-sequence is added to the video sequence.~~

~~14~~11. A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **c h a r a c t e r i z e d** in that the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and to

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format. ~~and to~~

12. The video encoder according to claim 11, wherein the video encoder is arranged to

form into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

13. The video encoder according to claim 11 or 12, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

14. The video encoder according to any of the claims 11 - 13, wherein the video encoder is arranged to

include the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI).

15. The video encoder according to any of the claims 11 - 14, **characterized** in that the video encoder is arranged to

encode the second sub-sequence to be removable from the video sequence without affecting decodability of the remaining video sequence.

16. A method for controlling bit rate of a video sequence transmission in a network element of a telecommunications system, **characterized** by

receiving, in the network element, a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

obtaining information indicative of dependencies of sub-sequences in the received video sequence;

controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

17. The method according to claim 16, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

18. A method according to claim 16 or 17, **characterized** by comprising

obtaining the information indicative of the sub-sequence from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI).

19. A method according to any of the claims 16 - 18, **characterized** by comprising

deducing on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

removing the second sub-sequence.

20. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to receive a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

the element is arranged to obtain information indicative of dependencies of sub-sequences in the received video sequence; and

the element is arranged to control the bit rate of at least one transmission of said video sequence on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

at least one sub-sequence is removed from the video sequence,

at least one sub-sequence is added to the video sequence.

21. The element according to claim 20, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

22. The element according to claim 20 or 21, **characterized** in that the element is arranged to

obtain the information indicative of the sub-sequence from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI).

23. The element according to any of the claims 20 - 22, **characterized** in that the element is arranged to

deduce on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

remove the second sub-sequence.

~~15. A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising the steps of~~

~~decoding a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;~~

~~decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;~~

~~determining dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and~~

~~reconstructing at least part of the video sequence on the basis of the dependencies of the sub-sequences.~~

~~16. A method according to claim 15, wherein video frames are entered into a sliding buffer memory in connection with decoding, **characterized** by comprising the steps of~~

decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;

configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

17. A method according to claim 16, **characterized** by comprising the step of

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

18. A method according to any one of claims 15 to 17, **characterized** by comprising the step of

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

19. A method according to any one claims 15 to 18, **characterized** by comprising the steps of

initiating the decoding from a random point in the video sequence;

determining an independently decodable sub-sequence that is next after the random point, and

setting the value of the image number of the first video frame in the sub-sequence at zero.

20. A method according to any one claims 15 to 19, **characterized** by comprising the steps of

identifying at least partly temporally overlapping sub-sequence frames from the video sequence based on the picture order count information, and

outputting from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.

21. A video decoder for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the

~~other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video decoder is arranged to~~

~~decode a first sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the first frame format;~~

~~decode at least a second sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;~~

~~determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and~~

~~reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.~~

~~22~~24. A computer program executable in a data processor for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format, ~~and~~

~~a program code for determining data into the video sequence to determine which video frames belong to the second sub-sequence.~~

25. A computer program executable in a data processor for controlling bit rate of a video sequence transmission, **characterized** in that the computer program comprises

a program code for obtaining information indicative of dependencies of sub-sequences in a received video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format; and

a program code for controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

~~23. A computer program for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the computer program comprises~~

~~a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and~~

a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.

24. A video signal comprising a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, characterized in that the video signal comprises

a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format;

at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

at least one data field determining the video frames that belong to the second sub-sequence.

MAIN REQUEST:

CLAIMS

1. A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising the steps of

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence;

wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

2. A method according to claim 1, **characterized** by

forming into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

3. A method according to claim 1 or 2, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

4. A method according to any preceding claim, **characterized** by comprising the step of

including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used

for transferring the video sequence, or as Supplemental Enhancement Information (SEI).

5. A method according to any preceding claim, **characterized** in that

the second sub-sequence being removable from the video sequence without affecting decodability of the remaining video sequence.

6. A method according to any preceding claim, **characterized** by comprising the steps of

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

7. A method according to claim 6, **characterized** by comprising the steps of

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from

a video frame of the scalability layer containing the first sub-sequence, and/or

a video frame of another sub-sequence in the same scalability layer.

8. A method according to claim 7, **characterized** by comprising the step of

determining the dependencies of the video frames of the second sub-sequence on the basis of at least the scalability layer identifier and the sub-sequence identifier.

9. A method according to claim 7 or 8, **characterized** by comprising the step of

coding a first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

10. A method according to claim 7 or 8, **characterized** by comprising the step of

coding a first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

.

11. A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and to

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

12. The video encoder according to claim 11, wherein the video encoder is arranged to

form into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

13. The video encoder according to claim 11 or 12, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

14. The video encoder according to any of the claims 11 - 13, wherein the video encoder is arranged to

include the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI).

15. The video encoder according to any of the claims 11 - 14, **characterized** in that the video encoder is arranged to

encode the second sub-sequence to be removable from the video sequence without affecting decodability of the remaining video sequence.

16. A method for controlling bit rate of a video sequence transmission in a network element of a telecommunications system, **characterized** by

receiving, in the network element, a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

obtaining information indicative of dependencies of sub-sequences in the received video sequence;

controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

17. The method according to claim 16, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

18. A method according to claim 16 or 17, **characterized** by comprising

obtaining the information indicative of the sub-sequence from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI).

19. A method according to any of the claims 16 - 18, **characterized** by comprising

deducing on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

removing the second sub-sequence.

20. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to receive a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

the element is arranged to obtain information indicative of dependencies of sub-sequences in the received video sequence; and

the element is arranged to control the bit rate of at least one transmission of said video sequence on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

at least one sub-sequence is removed from the video sequence,

at least one sub-sequence is added to the video sequence.

21. The element according to claim 20, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

22. The element according to claim 20 or 21, **characterized** in that the element is arranged to

obtain the information indicative of the sub-sequence from a header field of the video sequence or from a header field of a transfer protocol to be

used for transferring the video sequence, or from Supplemental Enhancement Information (SEI).

23. The element according to any of the claims 20 - 22, **characterized** in that the element is arranged to

deduce on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

remove the second sub-sequence.

24. A computer program executable in a data processor for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

25. A computer program executable in a data processor for controlling bit rate of a video sequence transmission, **characterized** in that the computer program comprises

a program code for obtaining information indicative of dependencies of sub-sequences in a received video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at

least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format; and

a program code for controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

2ND AUXILIARY REQUEST:

CLAIMS

1. A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising the steps of

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier, the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

2. A method according to claim 1, **characterized** by

forming into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

3. A method according to claim 1 or 2, wherein

the second sub-sequence being removable from the video sequence without affecting decodability of the remaining video sequence.

4. A method according to any preceding claim, **characterized** by comprising the steps of

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

5. A method according to claim 4, **characterized** by comprising the steps of

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from

a video frame of the scalability layer containing the first sub-sequence, and/or

a video frame of another sub-sequence in the same scalability layer.

6. A method according to claim 5, **characterized** by comprising the step of

determining the dependencies of the video frames of the second sub-sequence on the basis of at least the scalability layer identifier and the sub-sequence identifier.

7. A method according to claim 5 or 6, **characterized** by comprising the step of

coding a first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

8. A method according to claim 5 or 6, **characterized** by comprising the step of

coding a first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

9. A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence;

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

include the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier, the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

10. The video encoder according to claim 9, wherein the video encoder is arranged to

form into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

11. The video encoder according to claim 10 or 11, wherein the video encoder is arranged to

encode the second sub-sequence to be removable from the video sequence without affecting decodability of the remaining video sequence.

12. A method for controlling bit rate of a video sequence transmission in a network element of a telecommunications system, **characterized** by

receiving, in the network element, a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been

formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

obtaining information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of the received video sequence, the information indicative of the sub-sequence comprising a scalability layer identifier and a sub-sequence identifier;

controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

13. The method according to claim 12, **characterized** by comprising

deducing on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

removing the second sub-sequence.

14. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to receive a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one

video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

the element is arranged to obtain information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of the received video sequence, the information indicative of the sub-sequence comprising a scalability layer identifier and a sub-sequence identifier; and

the element is arranged to control the bit rate of at least one transmission of said video sequence on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

at least one sub-sequence is removed from the video sequence,

at least one sub-sequence is added to the video sequence.

15. The element according to claim 14, **characterized** in that the element is arranged to

deduce on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and remove the second sub-sequence.

16. A computer program executable in a data processor for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence;

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier, the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

17. A computer program executable in a data processor for controlling bit rate of a video sequence transmission, **characterized** in that the computer program comprises

a program code for obtaining information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of a received video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format, wherein the information indicative of the sub-

sequence comprises a scalability layer identifier and a sub-sequence identifier; and

a program code for controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

1ST AUXILIARY REQUEST:

CLAIMS

1. A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising the steps of

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

2. A method according to claim 1, **characterized** by

forming into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

3. A method according to claim 1 or 2, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

4. A method according to any preceding claim, **characterized** in that

the second sub-sequence being removable from the video sequence without affecting decodability of the remaining video sequence.

5. A method according to any preceding claim, **characterized** by comprising the steps of

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

6. A method according to claim 5, **characterized** by comprising the steps of

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from

a video frame of the scalability layer containing the first sub-sequence, and/or

a video frame of another sub-sequence in the same scalability layer.

7. A method according to claim 6, **characterized** by comprising the step of

determining the dependencies of the video frames of the second sub-sequence on the basis of at least the scalability layer identifier and the sub-sequence identifier.

8. A method according to claim 6 or 7, **characterized** by comprising the step of

coding a first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

9. A method according to claim 6 or 7, **characterized** by comprising the step of

coding a first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

.

10. A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence;

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

include the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

11. The video encoder according to claim 10, wherein the video encoder is arranged to

form into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence.

12. The video encoder according to claim 10 or 11, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

13. The video encoder according to any of the claims 10 - 12, **characterized** in that the video encoder is arranged to

encode the second sub-sequence to be removable from the video sequence without affecting decodability of the remaining video sequence.

14. A method for controlling bit rate of a video sequence transmission in a network element of a telecommunications system, **characterized** by

receiving, in the network element, a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

obtaining information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of the received video sequence;

controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

15. The method according to claim 14, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

16. A method according to claim 14 or 15, **characterized** by comprising

deducing on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

removing the second sub-sequence.

17. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to

transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to receive a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format;

the element is arranged to obtain information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of the received video sequence; and

the element is arranged to control the bit rate of at least one transmission of said video sequence on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

at least one sub-sequence is removed from the video sequence,

at least one sub-sequence is added to the video sequence.

18. The element according to claim 17, wherein the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier.

19. The element according to claim 17 or 18, **characterized** in that the element is arranged to

deduce on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and

remove the second sub-sequence.

20. A computer program executable in a data processor for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence;

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI);

wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

21. A computer program executable in a data processor for controlling bit rate of a video sequence transmission, **characterized** in that the computer program comprises

a program code for obtaining information indicative of dependencies of sub-sequences from a header field of the video sequence or from a header field of a transfer protocol to be used for transferring the video sequence, or from Supplemental Enhancement Information (SEI) of a received video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video

frame of the first sub-sequence being associated with information indicative of the first sub-sequence and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format; and

a program code for controlling the bit rate of at least one transmission of said video sequence from said network element on the basis of the information indicative of dependencies of sub-sequences in at least one of the following ways:

removing at least one sub-sequence from the video sequence,

adding at least one sub-sequence to the video sequence.

Europäisches Patentamt
European Patent Office
Office européen des brevets

Letter accompanying subsequently filed items

Sender:

Mr. Timo Pursiainen
Tampereen Patenttitoimisto Oy
Hermiankatu 1 B
33720 Tampere
Finland

Phone: +358 10 227 2600
Fax: +358 10 227 2662
E-mail: tampat@tampatent.eu

80298 Munich
Germany
Tel. +49(0)89 2399-0 | Fax -4465

P.O. Box 5818
NL-2280 HV Rijswijk
Netherlands
Tel. +31(0)70 340-2040 | Fax -3016

10958 Berlin
Germany
Tel. +49(0)30 25901-0 | Fax -840

The document(s) listed below is (are) subsequently filed documents pertaining to the following application:

Application number	03700321.7
Applicant's or representative's reference	TP103660/EPG

	Description of document	Original file name	Assigned file name
1	Reply to examination report	OA 20100602 response to Patent Office TP103660.pdf	EXRE3-1.pdf
2	Amended claims	OA 20100602 main request amended claims TP103660.pdf	CLMSPAMD-1.pdf
3	Amended claims	OA 20100602 1st aux request amended claims TP10366.pdf	CLMSPAMD-2.pdf
4	Amended claims	OA 20100602 2nd aux request amended claims TP10366.pdf	CLMSPAMD-3.pdf
5	Amended claims	OA 20100602 main request amended claims M-U TP1036.pdf	CLMSPAMD-4.pdf
6	Amended claims	OA 20100602 1st aux request amended claims M-U TP1.pdf	CLMSPAMD-5.pdf
7	Amended claims	OA 20100602 2nd aux request amended claims M-U TP1.pdf	CLMSPAMD-6.pdf

Signatures

Place:	**Tampere**
Date:	**14 October 2010**
Signed by:	**FI, PURSIAINEN TIMO 10009960X**
Capacity:	**(Representative)**

TAMPEREEN PATENTTITOIMISTO

13 October 2010

European Patent Office
Directorate General 2
D-80298 MÜNCHEN
Germany

VIA EPOLINE

Our ref.: TP103660/EPG

Response to the Summons to attend oral proceedings pursuant to Rule 115(1) EPC
European Patent Application No. 03700321.7 - 2223
Applicant: NOKIA CORPORATION

Dear Sirs,

This letter is in response to the summons to attend oral proceedings, dated 2 June 2010.

We respectfully request that the oral proceedings be suspended and the examination continued in writing on the basis of the main and auxiliary requests filed herewith. To this end we would welcome an opportunity to discuss this application by telephone with the Examiner, if considered necessary.

Please find enclosed three sets of claims, i.e. the main request and the first and second auxiliary request, which are submitted for further consideration for approval of the application.

1. The main request

As the main request, the Applicant files a new set of claims 1 – 25 to replace the claims 1 - 24 presently on file.

1.1. Multiple independent claims (A. 84, R. 43(2) EPC)

The idea underlying the present invention is to encode the dependencies of the sub-sequences within the video sequence by including information of all sub-sequences that have been directly used for predicting the image frames comprised by the sub-sequence in question. This information is usually signalled along the video sequence, preferably separate from the actual image information, so as to enable a streaming server or a corresponding network element to adjust the bit rate of the encoded video sequence by removing or adding sub-sequences, preferably without decoding the encoded video sequence.

Therefore, the inter-related operations are the encoding and the bit rate control rather than the encoding-decoding presented in the claims as filed. Accordingly, the set of amended claims include independent claim pairs for encoding-bit rate control method (claims 1 and 16), encoder-bit rate control element (claims 11 and 20) and encoding-bit rate control computer program (claims 24 and 25). The decoding-related claims have been deleted, as well as the video signal claim.

It is respectfully submitted that the streaming server, as disclosed e.g. in Fig. 1, does not typically carry out encoding operations, but only forwards the multimedia flows received from the editor to the clients, and at the same time, preferably carries out traffic shaping of the multimedia data, as described in paragraph [0028] of the present application.

1.2. Claim amendments

Thus, the new independent claim 1 discloses (amended features underlined, deleted ~~strikethrough~~): a method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames,

and the video frames of the second frame format being predicted from at least one of the other video frames, characterized by comprising the steps of

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence (*paragr [0034]; e.g. in Fig. 2, I-frame 200 is identified by the sub-sequence number 0*; *paragr. [0031]; Intra layer is assigned an identifier, such as the layer number 0)*; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence (*paragr [0036]; e.g. in Fig. 2, P-frame 204 is identified by the sub-sequence number 0 and its base layer is assigned the layer number 1)*, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence;

wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format (*paragr. [0032]; the sub-sequences of each scalability layer are assigned a unique identifier, paragr. [0031]; each scalability layer is assigned a unique identifier, such as the layer number*).

~~and~~

~~forming into the video sequence information indicating which video frames belong to the second sub-sequence.~~

The dependent claim 2 discloses: forming into the video sequence information indicating which video frames belong to the first sub-sequence and which video frames belong to the second sub-sequence (*paragr [0035 - 41]; e.g. in Fig. 2, together with the sub-sequence numbers and the scalability layer numbers, the consecutive image frame numbers identify the video frames belonging to each sub-sequence; otherwise*

the subject-matter has been shifted in a dependent claim, since the amended independent claim discloses how the sub-sequence information is defined).

The dependent claim 3 discloses: the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier (*paragr. [0032]; the sub-sequences of each scalability layer are assigned a unique identifier, paragr. [0031]; each scalability layer is assigned a unique identifier, such as the layer number; also claim 4 as filed*).

The dependent claim 4 discloses: including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI) (*paragr. [0085], also claim 9 as filed*).

The dependent claim 5 discloses: the second sub-sequence being removable from the video sequence without affecting decodability of the remaining video sequence (*paragr. [0017, 0033]*.

The dependent claim 7 is otherwise similar to the claim 3 as filed, but the possibly unclear definition “higher scalability layer” has been amended as “the scalability layer containing the first sub-sequence”.

The dependent claims 6 and 8 – 10 correspond to the claims 2 and 4 – 6 as filed.

Claims 11 – 15 and 24 define a video encoder and a computer program by features corresponding to claims 1 – 5.

Claim 20 (basis from the claim 13 as filed) discloses: An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to transmit compressed video sequences to at least

one terminal device and to control the bit rate of the video sequence transmission, characterized in that

the element is arranged to receive (*paragr. [0028]; multimedia files are transferred to a streaming server*) a video sequence comprising a first sub-sequence, at least part of which has been formed by coding video frames of at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence (*c.f. claim 1 above*), and at least a second sub-sequence, at least part of which has been formed by coding at least video frames of a second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence (*c.f. claim 1 above*), and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format (*c.f. claim 1 above*);

the element is arranged to obtain information indicative of dependencies of sub-sequences in the received video sequence (*paragr. [0020]; the streaming server is capable of deducing the inter-dependencies of the different sub-sequences directly from the identifiers and their inter-dependencies)*; and

the element is arranged to control the bit rate of at least one transmission of said video sequence on the basis of the information indicative of dependencies of sub-sequences (*paragr. [0083]; on the basis of this information, the streaming server decides how many scalability layers and which sub-sequences are transmitted in the video sequence*) in at least one of the following ways:

at least one sub-sequence is removed from the video sequence,

at least one sub-sequence is added to the video sequence.

The dependent claims 21 and 22 define the features corresponding to claims 13 and 14 (as well as claims 3 and 4).

The dependent claim 23 discloses: the element is arranged to deduce on the basis of the information indicative of the sub-sequence that the second sub-sequence is removable from the video sequence without affecting decodability of the remaining video sequence; and remove the second sub-sequence (*paragr. [0020]; the streaming server is capable of deducing the inter-dependencies of the different sub-sequences directly from the identifiers and their inter-dependencies; paragr. [0017]; independently decodable portions of the video sequence can be easily determined, and the portions can be removed without affecting the decoding of the rest of the image data)*.

Claims 16 – 19 and 25 define a method for controlling bit rate and a computer program by features corresponding to claims 20 – 23.

Thus, all the features of the new claims are disclosed in the application as filed. Hence, none of the amendments violates the Art. 123(2) EPC.

1.3. Novelty (A. 54 EPC)

The claims as filed were objected by Examiner due to lack of novelty in view document D1: Multimedia Signal Processing, 1998 IEEE Second Workshop on 7-9 Dec 1998, S. Wenger, "Temporal scalability using P-pictures for low-latency applications" pp. 559-564. The Examiner also submitted that the well-known MPEG coding resulting in a MPEG sequence comprising I and P frames was novelty-destroying prior art for the claims as filed.

D1 discloses a method for using P-pictures, instead of B-pictures, on the enhancement layers of a scalable video stream. The Examiner has interpreted the different layers as sub-sequences in the meaning of claim 1, the layer being identified by the picture type, and the information of the used reference picture signalled in the bit stream providing information about the dependencies of the sub-sequences (i.e. the layers).

However, D1 does not disclose that the information indicative of the sub-sequence is independent of the frame format. On the contrary, the argumentation of the Examiner is based on the idea that the sub-sequences (i.e. the layers) are identified by their picture types (the base layer by I/P-type and the enhancement layers by P-type).

In the similar manner, the MPEG-coded video sequence comprising I and P frames lacks the feature that the information indicative of the sub-sequence is independent of the frame format.

The same analogy applies to the independent claims 11 and 24 as well.

Regarding the bit rate control in a streaming server-type network element, wherein sub-sequences are removed from or added to an encoded video sequence, D1 is completely silent about such features. D1 only discusses about the increase in the bit rate caused in the encoding phase compared to using B-pictures on enhancement layers.

Thus, the independent claims 1, 11, 16, 20, 24 and 25 are novel in sense of Art. 54(1) EPC.

1.4. Inventive step (A. 56 EPC)

D1 discloses a method for using P-pictures, instead of B-pictures, on the enhancement layers of a scalable video stream. While causing a minor increase in the bit rate, the method enables to minimize latency in the decoding phase compared to using B-pictures on enhancement layers.

However, D1 gives no indication to encoding sub-sequences within scalability layers such that the bit rate of the video sequence could be adjusted in a streaming server-type network element by removing or adding sub-sequences without video sequence decoding being required. On the contrary, D1 emphasizes that scalability can only be achieved by removing

an entire layer (p. 5, last chapter) or if applied e.g. to MPEG coding, the complete video sequence must first be decoded before any adjustment of coding parameters can be done, and thereafter the video sequence must be re-encoded (p. 6, l. 6 – 8). This is evident also in view of the argumentation of the Examiner, where the layers are considered to be equivalent to sub-sequences, but the layers can only be identified their frame format type, i.e. no sub-sequences within scalability layers can be identified.

The characterising features of the amended claim 1 provide the technical effect of including in the video sequence information of independently decodable portions within scalability layers of of the video sequence, which portions can be removed without affecting decodability of the remaining frames of the scalability layer. Thus, the resulting overall technical problem to be solved is to create a coding hierarchy that enables to define such independently decodable portions within scalability layers and to indicate them along the encoded video sequence.

The problem is solved by a method comprising:
A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, characterized by comprising the steps of
forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format, at least one video frame of the first sub-sequence being associated with information indicative of the first sub-sequence; and
forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence being associated with information indicative of the second sub-sequence, and at least one video frame of the second sub-sequence has been

predicted from at least one video frame of the first sub-sequence;
wherein the information indicative of the first sub-sequence and the information indicative of the second sub-sequence are different from each other, and the information indicative of the sub-sequence is independent of the frame format.

Thus, by using such a flexible coding hierarchy it is enabled that, on one hand, the bit rate of a video sequence to be transmitted can be adjusted by removing sub-sequences even within scalability layers without video sequence decoding being required, and, on the other hand, independent decoding of each sub-sequence. This allows the streaming server, for example, to adjust the bit rate advantageously, without video sequence decoding, since the streaming server is capable of deducing the dependencies of the different sub-sequences directly from the identifiers and their dependencies.

D1 would not teach a skilled person to arrive at the claimed invention. On the contrary, D1 teaches that scalability can only be achieved layer-specifically, and D1 is completely silent about independently decodable sub-sequences within layers. Even less D1 gives indication to providing information about such sub-sequences to a steaming element in the network for adjusting the bit rate sub-sequence-specifically.

In the similar manner, a skilled person would not receive any indication for the claimed method from MPEG coding either.

Further, as submitted in D1, if the solution of D1 is applied to MPEG coding, the complete video sequence must first be decoded before any adjustment of coding parameters can be done, and thereafter the video sequence must be re-encoded. Accordingly, even a combination of D1 and D2 would not teach a skilled man to arrive at the claimed solution.

Consequently, the independent claims of the main request involve an inventive step in sense of Art. 56 EPC.

2. First auxiliary request

In comparison to the independent claims of the main request, the independent claims of the 1st auxiliary request include a further definition that including the information indicative of the sub-sequence in a header field of the video sequence or in a header field of a transfer protocol to be used for transferring the video sequence, or as Supplemental Enhancement Information (SEI) (*i.e. the subject-matter of the dependent claim 4 of the main request, paragr. [0085]*).

2.1. Inventive step (A.56 EPC)

The further definition included in the independent claims of the 1st auxiliary request provide the further technical effect that it defines more specifically how the information indicative of sub-sequence is attached to video sequence such that said information can be deduced without decoding the video sequence.

D1 gives no indication such solution. In addition to lacking the teaching towards information indicative of independently decodable sub-sequences within layers, D1 gives no solution for attaching such information. The only piece of information discussed in D1 to be transferred along the video sequence is the information about the used reference pictures, which is signalled in the picture header, i.e. the video sequence must be decoded at least to picture level in order to deduce said information. Thus, a skilled man would not receive any indication from to D1 to arrive at the claimed solution.

Consequently, the solution according to the independent claims of the 1st auxiliary request involves an inventive step.

3. Second auxiliary request

In comparison to the independent claims of the 1st auxiliary request, the independent claims of the 2nd auxiliary request

include a further definition that the information indicative of the sub-sequence comprises a scalability layer identifier and a sub-sequence identifier (*i.e. the subject-matter of the dependent claim 3 of the main and the 1st aux request; paragr. [0031-32])*

3.1. Inventive step (A.56 EPC)

The further definition included in the independent claims of the 1st auxiliary request provide the further technical effect that it defines more specifically the information elements, which can be used to identify independently decodable sub-sequences within scalability layers.

As discussed above, D1 gives no hints to independently decodable sub-sequences within layers, or to providing information about such sub-sequences to a steaming element. Thus, a skilled man would not receive any hints from D1 about the information elements, which can be used to identify independently decodable sub-sequences within scalability layers.

Consequently, the solution according to the independent claims of the 2nd auxiliary request involves an inventive step.

The following persons are announced to participate in the oral proceedings: European patent attorney Timo Pursiainen and patent attorney Eero Palmgren as attorneys familiar with the application, both from Tampereen Patenttitoimisto Oy, and the inventor, Mr. Miska Hannuksela.

With reference to the above request to suspend the oral proceedings, it is naturally in the interest of the Applicant to minimize or preferably avoid the travel costs caused by attending the oral proceedings. Therefore, if it appears to the Examiner that one or more of the submitted claim set would be allowable without a need for the oral proceedings, the Examiner is kindly requested to contact the undersigned (tel. +358-10-

2272 612) or patent attorney Eero Palmgren (tel. +358-10-2272 619) to discuss the matter further.

Yours faithfully,

Timo Pursiainen
Professional representative
Tampereen Patenttitoimisto Oy

ENCL. The main request
1st Auxiliary request
2nd Auxiliary request
Pages with amendments

Bitte umgehend zurücksenden!
Please return immediately!
Prière de renvoyer immédiatement !

EPO - Munich
46
11. Juni 2010

Empfangsbescheinigung über den Zugang des nachstehend bezeichneten Schriftstücks
Achnowledgement of receipt of the document specified below
Récépissé du document spécifié ci-après

Anmeldenummer / Beschwerdeaktenzeichen Application number / appeal case number N° de la demande / n° de référence du recours	Datum Date	EPA/EPO/OEB Form
03 700 321.7 (2223)	02.06.10	2008.1

Empfangen am / Received on / Reçu le

8 June 2010

Unterschrift / Signature

Katia Cerola

Empfangsberechtigter / authorised recipient /
le destinataire ou la personne dûment mandatée

EPA/EPO/OEB Form 2936 12.06

Zur gefälligen Beachtung
Unter Bezugnahme auf die Mitteilung im ABl. des EPA 11/1991, 577, wird gebeten, die Empfangsbescheinigung mit **Empfangsdatum** und **Unterschrift** zu versehen und **umgehend** an das EPA zurückzusenden.

Please note
With reference to the Notice in EPO OJ 11/1991, 577, you are requested to **date** and **sign** the acknowledgement of receipt an return it to the EPO **immediately**.

Nota bene
Conformément à la communication parue au JO OEB 11/1991, 577, vous êtes prié de porter sur le récépissé la **date d'arrivée** du document, ainsi que votre **signature** et de le renvoyer **immédiatement** à l'OEB.

An das

Europäische Patentamt

80298 München

ALLEMAGNE / GERMANY / DEUTSCHLAND

The examination is being carried out on the **following application documents**

Description, Pages

1-34 as published

Claims, Numbers

1-24 as published

Drawings, Sheets

1/4-4/4 as published

1 Oral proceedings will take place in accordance with Article 116(1) EPC at the applicants request and because the Examining Division considers it to be expedient.

2 In the official communication of 16.06.2006, objections were raised with respect to multiple independent claims in each category (Rule 43(2) EPC, previous Rule 29(2) EPC), lack of clarity (Article 84 EPC) and lack of novelty (Article 52(1) and 54(1) and (2) EPC).

3 In a letter of 16.10.2006, the applicants provided arguments why the present application meets the requirements of the EPC. The claims were left unamended.

4 Having carefully considered the applicant's arguments, the Examining Division is of the opinion that the present application does not meet the requirements of the EPC and that for the following reasons:

5 Multiple independent claims (Article 84, Rule 43(2) EPC)

The present application contains 8 independent claims, namely claim 1 related to a method for coding, claim 13 related to an element in a streaming system, claim 14 related to a video encoder, claim 15 related to a method for decoding, claim 21 related to a video decoder, claim 22 related to a computer program for coding, claim 23 related to a computer program for decoding and claim 24 related to a video signal.

Under Rule 43(2) EPC, an application may not contain more than one independent claim in a particular category unless the subject-matter claimed falls within one or more of the exceptional situations set out in paragraph (a), (b) or (c) of Rule 43(2) EPC.

The applicant points out that a number of claims are interrelated in that they represent encoder-decoder type inventions, which are an exception to the general rule (Rule 43(2)(a) EPC). The encoder-decoder pairs are:

- claims 1 and 15 (encoding-decoding method);
- claims 14 and 21 (encoder-decoder); and
- claims 22 and 23 (encoding-decoding computer program)

The Examining Division agrees that separate independent claims to an encoder and the corresponding decoder are allowed. However, the encoder-decoder claims in the present case are not strictly interrelated, because the decoders do not reverse the steps of the encoders. In particular, the decoder claims do not refer to dependency data **present in the bit stream**. This also gives rise to a lack of clarity and support by the description (Article 84 EPC), because the dependency information, signalled in the bit stream, is clearly presented as an essential feature in the description (see paras. 0016 and 0033).

Claim 13, which defines an element in a streaming system arranged to control the bit rate of a video sequence transmission, is, in effect, a claim to an encoder. Thus, there are two encoder claims, i.e. claims 13 and 14. The subject-matter of claim 13 is quite different from claim 14. However, both claims relate to the same sub-sequence structure and cannot, therefore, be regarded as alternative solutions to the same problem. Hence, the exception of Rule 43(2)(c) EPC does not apply.

In conclusion, the present set of claims contravenes Rule 43(2) EPC, which is a ground for refusal under Article 97(2) EPC. The Examining Division is bound by the EPC and not by the law of the contracting states or the "general viewpoint of the patenting system". The Examining Division can only adopt a "flexible approach" within the bounds of the EPC.

6 Lack of novelty (Article 52(1) and 54(1) and (2) EPC)

6.1 The present application does not meet the requirements of Article 52(1) EPC because the subject-matter of claims 1, 13-15 and 21-24 is not new within the meaning of Article 54(1) and (2) EPC.

6.2 Claim 1 refers to a method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, comprising forming a first sub-sequence, at least part of

which being coded according to the first format and a second sub-sequence, at least part of which being coded according the the second format, and at least one of the frames of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence, and forming into the video sequence information indicating which video frames belong to the second sub-sequence.

6.3 Thus, claim 1 includes the case where the video sequence is a normal MPEG sequence comprising I and P frames. The I frames represent the first sub-sequence (a single I-frame may form a separate sub-sequence, see para. 0035) and the P-frames the second sub-sequence. The picture-type information present in the picture header indicates the picture type (I or P), and thus implicitly which frames belong to the second sub-sequence. An MPEG sequence may be regarded as scalable in the sense that P-frames, i.e. the second sub-sequence, can be discarded. MPEG coding was indisputably well known at the priority date of the present application. Claim 1 is, therefore, not new within the meaning of Article 54(1) and (2), and the requirements of Article 52(1) are not satisfied.

6.4 Furthermore, claim 1 lacks novelty with respect to D1 (Article 54(1) and (2) EPC).

D1 is a document related to scalable coding using P-pictures (title). The base layer is a sequence of I or P pictures (fig. 2; page 3, line 16) and the enhancement layer consists of P pictures (fig. 2; page 3, line 17). The enhancement layer uses previous pictures of the enhancement layer as well as base layer pictures as reference pictures (fig. 2; page 3, lines 18-19 and fig. 2). Thus, the enhancement layer forms one or several separate sub-sequences (see fig. 2). In addition to the picture type information, which identifies the P-frames and consequently the enhancement layer in the case where the base layer consists of only I pictures), the reference picture used is signalled into the bit stream, thus providing information about the interdependencies of the respective sub-sequences (page 4, lines 6-8).

6.5 The applicant argues in his letter of reply that different temporal layers are not sub-sequences as disclosed in the present application. The Examining Division cannot follow this argument. The wording of claim 1 does not exclude this possibility. Furthermore, according to the description, a scalability layer, or even a single frame, may constitute a sub-sequence (see para. 0034).

The applicant cites the passage "The base layer is the only self-contained representation of the sequence [...]" in D1, page 1, as support for his argument that the temporal layers in D1 are not sub-sequences in the sense

of claim 1. However, the Examiner does not see any difference with respect to the present application. The second sub-sequence in claim 1, which consists of predicted frames, is not self-contained (see also fig. 2 of the present application).

6.6 The applicant also argues that D1 does not disclose information indicating which video frames belong to the second sub-sequence; it just indicates the reference picture. The Examining Division disagrees. The information in D1 constitutes dependency information, identifying the separate layers (sub-sequences). Thus, the bit stream information in D1 indicates, albeit implicitly, which frames belong to the second sub-sequence.

6.7 In conclusion, the applicant's arguments are not convincing. The Examining Division maintains that claim 1 lacks novelty with respect to D1. Also, the claim is so broad that it lacks novelty with respect to a normal MPEG sequence.

6.8 For the same reasons, claims 14, 15 and 21-24 lack novelty (Article 54(1) and (2) EPC). The requirements of Article 52(1) EPC are, therefore, not satisfied.

6.9 Furthermore, D1 discloses video transmission where the bit-rate is controlled by removing scalability layers (sub-sequences) from the bit stream (para. Introduction). The subject-matter of claim 13 is, therefore, not new either (Article 52(1) and 54(1) and (2) EPC).

7 The dependent claims do not appear to contain anything which contributes towards and inventive step.

8 Failure to overcome the above objections may lead to a refusal of the application under Article 97(2) EPC at the end of the oral proceedings.

***

Europäisches Patentamt
European Patent Office
Office européen des brevets

European Patent Office
80298 MUNICH
GERMANY
Tel. +49 (0)89 2399 - 0
Fax +49 (0)89 2399 - 4465

Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy
Hermiankatu 1 B
33720 Tampere
FINLANDE

Formalities Officer
Name: Simão
Tel.: 8637
or call:
+31 (0)70 340 45 00

Date
02-06-2010

Reference	Application No./Patent No.
TP103660	03700321.7 - 2223 / 1479245
Applicant/Proprietor	
Nokia Corporation	

Summons to attend oral proceedings pursuant to Rule 115(1) EPC

You are hereby summoned to attend oral proceedings arranged in connection with the above-mentioned European patent application.

The matters to be discussed are set out in the communication accompanying this summons (EPO Form 2906).

The oral proceedings, which will not be public, will take place before the Examining Division.

on 17.11.10 at 09.30 hrs at the EPO,
PschorrHöfe, Bayerstr. 34, D-80335 München

No changes to the date of the oral proceedings can be made, except on serious grounds (see OJ EPO 1/2009, 68). If you do not appear as summoned, the oral proceedings may continue without you (R. 115(2) EPC, see also OJ EPO 10/2008, 471).

Your attention is drawn to Rule 4 EPC, regarding the language of the oral proceedings, and to the Special edition No. 3 OJ EPO 2007, L.1., concerning the filing of authorisations for company employees and lawyers acting as representatives before the EPO.

The final date for making written submissions and/or amendments (R. 116 EPC) is 15.10.10.

The actual room number as well as the waiting room numbers will be given to you by the porter in the foyer at the above EPO address.

Parking is available free of charge in the underground car park. However, this applies only in the case of accessing the car park via the entrance "Zollstrasse".

1st Examiner:	2nd Examiner:	Chairman:
Wahrenberg I	Hindelang T	Willems B

For the Examining Division

Annexes:
Confirmation of receipt (Form 2936)
Communication (EPO Form 2906)

Europäisches Patentamt • European Patent Office • Office européen des brevets

Registered letter with advice of delivery

to EPO postal service: 27.05.10

Europäisches Patentamt
European Patent Office
Office européen des brevets

European Patent Office
80298 MUNICH
GERMANY
Tel: +49 89 2399 0
Fax: +49 89 2399 4465

Application No.: 03 700 321.7

Preparation for oral proceedings - Instructions to Support Service

Oral proceedings are to be held in connection with the above patent application

1. The matters to be discussed are set out in the annex (Form 2906)

2. Dispatch the summons using Form 2008/2310 and Form 2906 for the parties to attend on:

Day 17.11.2010 Time 09:30

ROOMS

Room booked

ORAL 01, 02, 03 and 05 coded

27.05.10 si
Date Initials

2.1 Parties' submissions in preparation for the oral proceedings, if any, should be made no later than

1 month(s)

before the date of the oral proceedings
(transfer to Form 2008.1 / 2310.1)

2.2 Encode ORAL(04) coded

27.05.10 si
Date Initials

2.3 Dispatch Form 2008.7 / 2310.7 to division

27.05.10 si
Date Initials

3. ☐ Arrange for the following special equipment to be provided in the conference room:

Date Initials

Europäisches Patentamt
European Patent Office
Office européen des brevets

European Patent Office
80298 MUNICH
GERMANY
Tel: +49 89 2399 0
Fax: +49 89 2399 4465

4. Request language service to provide simultaneous interpretation facilities as necessary

Date Initials

5. Return the dossier to primary examiner with Form 2041 (15 days before the oral proceedings)

Date Initials

6. Check that summons has been received (Form 2936 / advice of delivery)

7. 15 days before the oral proceedings:
- dispatch the dossier to the primary examiner.

30.4.2010
Date

Willems, Brigitte
Chairman

Hindelang, Thomas
2nd examiner

Wahrenberg, Annika
1st examiner

Legal member

Enclosure(s):

03700321.7 - 1247 / 1479245 05.08.08

Client Database System (CDS) - clean up.

Application Nr.: 03700321.7

Following clean up action in CDS the entries concerning the **Representative for the applicant** have been amended and are now as follows:

Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy
Hermiankatu 1 B
33720 Tampere
FI

Where appropriate, the Register of European Patents will be updated to show the amended details.

For questions please contact the Client Data Registration department of the European Patent Office in Munich, telephone +49 (0)89 2399 2780.

PATENTIT • PATENTS
TAVARAMERKIT • TRADEMARKS
MALLIT • DESIGNS
HYÖDYLLISYYSMALLIT • UTILITY MODELS

TAMPEREEN PATENTTITOIMISTO

OUR CONTACT INFORMATION HAS CHANGED

Tampereen Patenttitoimisto Oy has moved. We continue to serve you in all IPR matters in
Tampere, Finland. Our new address is:

EPO - Munich
62
06. März 2008
USt

Hermiankatu 1 B
FI-33720 Tampere
Finland

Our telephone and fax numbers have also changed. The new numbers are:

Telephone: +358 10 227 2600
Fax: +358 10 227 2662

There are no changes in our e-mail addresses:
firstname.lastname@tampatent.eu or
firstname.lastname@tampereenpatenttitoimisto.fi.

From 1 March 2008 you can also reach us in Lappeenranta. The contact information for our Lappeenranta office is:

from 1 March 2008 to 30 April 2008

Laserkatu 6
FI-53850 Lappeenranta
Finland

S FREP

from 1 May 2008 onward

Valtakatu 49
FI-53100 Lappeenranta
Finland

Telephone: +358 10 227 2634

E-Bild/Euc/UST

Yours sincerely,

Tampereen Patenttitoimisto Oy

www.berggren.fi

BERGGREN-YHTIÖT • BERGGREN GROUP

TAMPEREEN PATENTTITOIMISTO OY
Hermiankatu 12 B
FIN-33720 Tampere
FINLAND

KÄYNTIOSOITE • OFFICE
Hermia
Hermiankatu 12 B
Tampere

PUH. • TEL
Nat. (03) 288 6111
Int. +358 3 288 6111
Fax +358 3 288 6262

E-MAIL
tampat@
tampereenpatenttitoimisto.fi
www.berggren.fi

PANKIT • BANKERS
NORDEA 224318-2090
NORDEA USD 224362-2384
SWIFT NDEAFIHH
SAMPO 800019-733966

YHTIÖ • COMPANY
Y 0707748-1
VAT FI07077481
Kotipaikka Tampere
Domicile Tampere

Page 1946 of 2504

Posted

isches
mt

European Patent Office

Office européen des brevets

Einsender / Sender / Expéditeur :
(bitte ausfüllen / please fill / à remplir svp)

Mr Timo Pursiainen
Tampereen Patenttitoimisto Oy
Hermiankatu 12 B
FI-33720 TAMPERE
FINLAND

D-80298 München
(+49-89) 23 99-0
Fax (+49-89) 23 99-44 65

P.B. 5818 Patentlaan 2
NL-2280 HV Rijswijk
(+31-70) 340-20 40
Fax (+31-70) 340-30 16

D-10958 Berlin
(+49-30) 259 01-0
Fax (+49-89) 259 01-840

Bestätigung[2] über den Eingang nachgereichter Unterlagen für Patentanmeldungen/Patente beim Europäischen Patentamt

Datum und Ort des Eingangs sind aus dem Eingangsstempel bzw. der Perforation dieser Eingangsbestätigung ersichtlich.
(M + Datum = Einreichungsort München;
H + Datum = Einreichungsort Den Haag;
Datum + B = Einreichungsort Berlin)

Acknowledgement of receipt[2] for subsequently filed items relating to patent applications/ patents at the European Patent Office

Date and place of receipt are shown by the receipt stamp or perforation appearing on this receipt.
(M + date = Munich as place of receipt;
H + date = The Hague as place of receipt;
date + B = Berlin as place of receipt)

Accusé de réception[2] à l'Office européen des brevets de pièces produites postérieurement au dépôt d'une demande de brevet/ à la délivrance d'un brevet européen

La date et le lieu de réception sont indiqués par le cachet de réception ou la perforation du présent accusé de réception.
(M + date = pièces reçues à Munich;
H + date = pièces reçues à La Haye;
date + B = pièces reçues à Berlin)

Eingereichte Unterlagen / Items filed / Pièces envoyées

	Anmeldenummer/Patentnummer Application Number/Patent Number Numéro de la demande/numéro du brevet	Ihr Zeichen Your reference Votre référence	ggfs. Art und Datum der Unterlagen [3] Nature and date of items (optional) [3] Nature et date des pièces (facultatif) [3]
1.	03 700 321.7 - 1247	TP103660/EPG	Respose dated 16.10.2006
2.			
3.			
4.			
5.			
6.			
7.			Pages in total 7
8.			
9.			
10.			

1) bitte Adressfeld ausfüllen
2) bitte 2-fach einreichen
3) Der Eingang der angegebenen Unterlagen wird bestätigt. Enthält diese Spalte keine Eintragungen, so wird lediglich bestätigt, dass eine Sendung zu dem angegebenen Aktenzeichen eingegangen ist

1) please fill in address
2) please submit in 2 copies
3) The receipt of the items indicated is confirmed. If this column does not contain any entries, it is only confirmed that an item has been received for the indicated file

1) adresse à remplir, svp.
2) à soumettre en 2 exemplaires, svp
3) La réception des pièces indiquées est confirmée. Faute de mention dans cette colonne, le présent accusé de réception se rapporte à une pièce quelconque envoyée sous la référence indiquée

EPA/EPO/OEB Form 1037.1 02.06
Eingangsbestätigung für Einsender
Acknowledgement of receipt for sender
Accusé de réception expéditeur

Received at the EPO on Oct 16, 2006 14:47:01. Page 1 of 7

Europäisches Patentamt | European Patent Office | Office européen des brevets

Einsender / Sender / Expéditeur :
(bitte ausfüllen / please fill / à remplir svp)

Mr Timo Pursiainen
Tampereen Patenttitoimisto Oy
Hermiankatu 12 B
FI-33720 TAMPERE
FINLAND

D-80298 München
(+49-89) 23 99-0
Fax (+49-89) 23 99-44 65

P.B. 5818 Patentlaan 2
NL-2280 HV Rijswijk
(+31-70) 340-20 40
Fax (+31-70) 340-30 16

D-10958 Berlin
(+49-30) 259 01-0
Fax (+49-89) 259 01-840

Bestätigung[2] über den Eingang nachgereichter Unterlagen für Patentanmeldungen/Patente beim Europäischen Patentamt

Datum und Ort des Eingangs sind aus dem Eingangsstempel bzw. der Perforation dieser Eingangsbestätigung ersichtlich.
(M + Datum = Einreichungsort München;
H + Datum = Einreichungsort Den Haag;
Datum + B = Einreichungsort Berlin)

Acknowledgement of receipt[2] for subsequently filed items relating to patent applications/ patents at the European Patent Office

Date and place of receipt are shown by the receipt stamp or perforation appearing on this receipt.
(M + date = Munich as place of receipt;
H + date = The Hague as place of receipt;
date + B = Berlin as place of receipt)

Accusé de réception[2] à l'Office européen des brevets de pièces produites postérieurement au dépôt d'une demande de brevet/ à la délivrance d'un brevet européen

La date et le lieu de réception sont indiqués par le cachet de réception ou la perforation du présent accusé de réception.
(M + date = pièces reçues à Munich;
H + date = pièces reçues à La Haye;
date + B = pièces reçues à Berlin)

Eingereichte Unterlagen / Items filed / Pièces envoyées

	Anmeldenummer/Patentnummer Application Number/Patent Number Numéro de la demande/numéro du brevet	Ihr Zeichen Your reference Votre référence	ggfs. Art und Datum der Unterlagen [3] Nature and date of items (optional) [3] Nature et date des pièces (facultatif) [3]
1.	03 700 321.7 - 1247	TP103660/EPG	Respose dated 16.10.2006
2.			
3.			
4.			
5.			
6.			
7.			Pages in total 7
8.			
9.			
10.			

1) bitte Adressfeld ausfüllen
2) bitte 2-fach einreichen
3) Der Eingang der angegebenen Unterlagen wird bestätigt. Enthält diese Spalte keine Eintragungen, so wird lediglich bestätigt, dass eine Sendung zu dem angegebenen Aktenzeichen eingegangen ist.

1) please fill in address
2) please submit in 2 copies
3) The receipt of the items indicated is confirmed. If this column does not contain any entries, it is only confirmed that an item has been received for the indicated file.

1) adresse à remplir, svp.
2) à soumettre en 2 exemplaires, svp.
3) La réception des pièces indiquées est confirmée. Faute de mention dans cette colonne, le présent accusé de réception se rapporte à une pièce quelconque envoyée sous la référence indiquée

EPA/EPO/OEB Form 1037.2 02.06
Kopie für EPA
Copy for EPO
Exemplaire OEB

Received at the EPO on Oct 16, 2006 14:47:01. Page 2 of 7

PATENTIT • PATENTS
TAVARAMERKIT • TRADEMARKS
MALLIT • DESIGNS
HYÖDYLLISYYSMALLIT • UTILITY MODELS

TAMPEREEN PATENTTITOIMISTO

16 October 2006

VIA FACSIMILE

European Patent Office
Directorate General 2
D-80298 MÜNCHEN
Germany

Our Ref: TP103660 EPG

EUROPAN PATENT APPLICATION NO. 03 700 321.7 - 1247
APPLICANT: NOKIA CORPORATION

Dear Sirs

This letter is in response to the official communication dated 16.6.2006.

The Applicant believes that the claims in their current form satisfy the requirements of the patentability in the EPC, and the Applicant submits the following arguments:

Number of independent claims (A. 84, R.29(2) EPC)

Regarding the objection of more than one independent claim in the same category (R.29(2) EPC), the Applicant respectfully disagrees. The independent claims 1, 14 and 22 disclose a method for encoding a video sequence, an encoder and a computer program for encoding. The independent claims 15, 21 and 23, in turn, disclose a method for decoding a video sequence, an decoder and a computer program for decoding. These different aspects of the invention represent an encoder-decoder-type invention concerning a plurality of inter-related products having the same inventive concept, which are allowed according to R. 29(2)(a) EPC. Furthermore, the independent claim 13 discloses an element for controlling the bit rate of the video sequence and the independent claim 24 discloses a video signal encoded according to the method of claim 1. All these different aspects of the invention can be manufactured, delivered and carried out as separate products, entities or actions and in order for the invention to be fully covered, it is essential that a possible infringement against any aspect of the invention can be counteracted. Even though claims 13, 14 and 21 – 24 all define apparatuses, these claim categories are of vital importance for the Applicant in obtaining the full protection for his invention in all designated

EPC contracting states. With this basic viewpoint of the whole patenting system in mind, it is desirable that the Examiner would adopt a flexible approach when evaluating several independent claims in the same category.

Furthermore, from the wording of independent claims 1, 13 - 15 and 21 - 24, it is clear that they all share a unifying inventive (encoding-decoding) concept, whereby the requirement of unity according to Art. 82 EPC is also satisfied and the determination of the extent of the protection is not impeded by any means.

Clarity objections (R. 84 EPC)

The Examiner objects the indeonedent claims, especially claim 1, as being not clear and being inconsistent with various embodiments of the description. The Applicant respectfully disagrees.

Particularly, the Examiner's interpretation of the term "sequence" as implying consecutive frames and not different layers cannot be agreed with. Claim 1 is defined as "a method for coding [...] a scalable, compressed video sequence...". A skilled man readily appreciates that in a scalable video sequence there are typically a plurality of layers, each including their own video sequences. Moreover, in the chap. 0044 it is disclosed: "The sub-sequences of the same layer or of the different layer may be partly or entirely overlapping". Also, the Oxford English dictionary defines a sequence as "a set of related events, movements, or things that follow each other in a particular order"; i.e. by the definition, the frames do not have be consecutive.

A sub-sequence is defined in the application as "consisting of groups of one or more video frames determined for each scalability layer" (chap. 0032, first sentence). The other essential properties of a subsequence are determined in claim 1.

Referring to Figure 2, some examples of sub-sequence can be given: In INTRA layer, frame 200 is an example of "a first sub-sequence" (this sub-sequence only contains 1 frame); In Base layer, frames 204, 206 are an example of "a second sub-sequence" (this sub-sequence contains 2 frames), wherein frame 204 is predicted from a frame of the first sub-sequence (frame 200); In Enh1 layer, frame 208 is also an example of "a second sub-sequence" (this sub-sequence contains only 1 frame); Also in Enh1 layer, frames 210, 212

Received at the EPO on Oct 16, 2006 14:47:01. Page 4 of 7

are also an example of "a second sub-sequence" (this sub-sequence contains 2 frames). This example is further clarified in the corresponding part of by the detailed description of the invention.

Furthermore, the application discloses an identifier that can be used to identify the frames and sub-sequences. An example of such identifier is composed of an image type, a subsequence identifier, and a image number (chap. 0034). Furthermore, in chap. 0057 it is pointed out that "For clarity of illustration, the above description of the embodiment does not comprise layer identifiers, but it is apparent that in order to implement scalability, also the layer identifier must be signalled together with the video sequence, typically as part of the image frame identifiers."

Again, referring to figure 2, frame 200 in the INTRA layer is labeled as (I.0.0), which means this is an I frame, and it belongs to subsequence 0, and within subsequence 0, the image number is 0. Similarly, frames 204 and 206 in the Base layer are labelled as (P.0.0) and (P.0.1) respectively, which means that both are P frames, and both belong to subsequence 0, and the image number within the subsequence 0 is 0 and 1 respectively. Also, the frames 210, 212 in Enh1 layer are labeled as (P.1.0) and (P.1.1) respectively, meaning that both are P frames, and both belong to subsequence 1, and the image number within subsequence 1 is 0 and 1 respectively. Using the above example, the frames 210, 212 forms an example of "a second sub-sequence", wherein at least one video frame (210) has been predicted from at least one video frame (200) of the first subsequence. Thus, it can be concluded that the subject-matter of claim 1 is consistent with figure 2.

Novelty (A. 54 EPC)

Regarding the novelty objection raised by the Examiner in view of the document D1: Multimedia Signal Processing, 1998 IEEE Second Workshop on 7-9 Dec 1998, S. Wenger, "Temporal scalability using P-pictures for low-latency applications" pp. 559-564, it is respectfully submitted that said document does not disclose any sub-seqeunce in the meaning as it is disclosed in the present application.

However, even if the Examiner takes the liberty to interpret the different layers as sub-sequences in the meaning of claim 1, it is still submitted that D1 does not disclose that the video sequence includes information indicating which video frames belong to the second sub-sequence. The mere fact that

Received at the EPO on Oct 16, 2006 14:47:01. Page 5 of 7

information about a reference picture, which can be used as reference for further prediction, is indicated in the video stream does not disclose, which video frames belong to the second sub-sequence; it just tells that there is a reference picture, to which one or more later pictures can refer as a motion prediction basis. This is further underlined by the statement: "The base layer is the only self-contained representation of the sequence [...]" (D1, p. 1) .Thus, the allegation of lack of novelty is groundless and the claims 1, 13 - 15 and 21 – 24 are novel in sense of A.54(1),(2) EPC.

Inventive step (A. 56 EPC)

D1 discloses a method for using P-pictures, instead of B-pictures, on the enhancement layers of a scalable video stream. While causing a minor increase in the bit rate, the method enables to minimize latency in the decoding phase compared to using B-pictures on enhancement layers.

However, the solution of D1 includes technical drawbacks. Adjusting the scalabilty e.g. in a streaming server or a network element becomes difficult, since the complete video sequence must first be decoded before any adjustment of coding parameters can be done, and thereafter the video sequence must be re-encoded (D1, p. 6, l. 6 – 8). This also one of the drawbacks of the prior art mentioned in the present application (chap. 0011). D1 is also completely silent about frame numbering, but since it is commonly known in the prior art to use ascending arithmetical frame numbering, which most probably is used in the solution of D1 as well, although not expressly stated. Such a frame numbering, however, becomes problematic, when inserting a video sequence in the middle of another video sequence or scaling of a video sequence, as is disclosed in the present application (chap. 0013).

Thus, the objective technical problems underlying the present invention are the same as the drawbacks of the prior art mentioned in the present application: carrying out reference picture selection when the prediction includes images temporally succeeding the image to be coded, how to signal differences of significance between INTER-frames, and how to create a frame numbering of video sequence, which allows flexible the insertion of a video sequence in the middle of another video sequence and scaling of a video sequence.

The problem is solved by a method for coding video frames for the purpose of forming a scalable, compressed video sequence, wherein a video signal is formed, which video

Received at the EPO on Oct 16, 2006 14:47:01. Page 6 of 7

signal comprises a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format; at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and at least one data field determining the video frames that belong to the second sub-sequence.

Hence, the idea underlying the present invention is to create a scalable coding hierarchy, wherein, for each sub-sequence, all sub-sequences are determined, upon which the coding of said sub-sequence is dependent. This is achieved by a frame numbering hierarchy, wherein the frame numbering of each sub-sequence includes information on all sub-sequences, which have been used in image frame prediction. This information is signalled along the video sequence, but separately of the image information. This enables a flexible scaling of video sequence, since independently decodable parts of the video sequence can easily be determined.

D1 includes no hints to this kind of solution. D1 does not disclose any independently decodable sub-sequence on enhancement layers, as is disclosed in the present application. D1 does not disclose or even hint that the video sequence could include information indicating which video frames belong to the second sub-sequence, nor does it include any information about how the frame numbering of the video sequence should be carried out.

Therefore, claims 1, 13 - 15 and 21 – 24 involve an inventive step in sense of A.56 EPC.

In view of the amendments and submissions now made, the applicant respectfully submits that the application as a whole is now in order for allowance. If the Examiner is of the opinion that further objections should be raised, then a further written communication is requested. In the event that the Examiner considers any further objections cannot be dealt with in writing, then oral proceedings are requested in precaution under Article 116 EPC.

Yours faithfully

Timo Pursiainen, Authorised Representative
TAMPEREEN PATENTTITOIMISTO OY

Received at the EPO on Oct 16, 2006 14:47:01. Page 7 of 7

P.B.5818 - Patentlaan 2
2280 HV Rijswijk (ZH)
(070) 3 40 20 40
FAX (070) 3 40 30 16

Europäisches Patentamt | **European Patent Office** | **Office européen des brevets**

Generaldirektion 1 | Directorate General 1 | Direction générale 1

Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy,
Hermiankatu 12 B
33720 Tampere
FINLANDE

EPO Customer Services
Tel.: +31 (0)70 340 45 00

Date 15-08-2006

Reference	Application No./Patent No.
TP103660	03700321.7 - 1247

Applicant/Proprietor
Nokia Corporation

BRIEF COMMUNICATION

Subject:
- [x] Your letter of 07-07-2006
- [ ] Our telephone conversation of
- [ ] Communication of
- [ ]

Enclosure(s):
- [ ] Letter from the proprietor of the patent of
- [ ] Letter from the opponent of
- [x] Copy(copies) missing cited document
- [ ] Communication:

We refer to your fax of 07-07-2006.
We hereby send you a copy of the missing cited document : Wenger - Temporal scalabilty using P-picture for low-latency application.
We apologize for the delay and the inconvenience caused.

Please take note.

For the Examining Divison

Iacoponi, Cristina

Europäisches Patentamt • European Patent Office • Office européen des brevets

☐ **Registered letter**

EPO Form 2911 05.02 10.08.06

CI02253

PATENTIT • PATENTS
TAVARAMERKIT • TRADEMARKS
MALLIT • DESIGNS
HYÖDYLLISYYSMALLIT • UTILITY MODELS

7 July 2006
VIA FACSIMILE

TAMPEREEN PATENTTITOIMISTO

European Patent Office
Directorate General 2
D-80298 MÜNCHEN
Germany

Our ref.: TP103660 EPG/SPU

RE: EUROPEAN PATENT APPLICATION NO. 03700321.7-1247
in the name of NOKIA CORPORATION
Communication pursuant to Article 96(2) EPC

Dear Sirs,

With reference to the Communication pursuant to Article 96(2) EPC dated 16 June 2006, please be informed that the cited document (*Wenger: Temporal scalability using P-picture for low-latency applications*) was not annexed to the received communication. Therefore, we request that a copy of the document be sent to us as soon as possible.

Yours very truly,

TAMPEREEN PATENTTITOIMISTO OY

Erkki Rahkonen
Representative

BERGGREN-YHTIÖT • BERGGREN GROUP

TAMPEREEN PATENTTITOIMISTO OY
Hermiankatu 12 B
FIN-33720 Tampere
FINLAND

KÄYNTIOSOITE • OFFICE
Hermia
Hermiankatu 12 B
Tampere

PUH. • TEL
Nat. (03) 268 6111
Int. +358 3 268 6111
Fax +358 3 268 6262

E-MAIL
tampat@
tampereenpatenttitoimisto.fi
www.berggren.fi

PANKIT • BANKERS
NORDEA 224318-2090
NORDEA USD 224362-2384
SWIFT NDEAFIHH
SAMPO 800019-733986
SWIFT PSPBFIHH

YHTIÖ • COMPANY
Y 0707748-1
VAT FI07077481
Kotipaikka Tampere
Domicile Tampere

Received at the EPO on Jul 07, 2006 10:09:23. Page 1 of 1

The examination is being carried out on the **following application documents**:

Description, Pages

1-34 as published

Claims, Numbers

1-24 as published

Drawings, Sheets

1/4-4/4 as published

1 The following documents is cited by the examiner (see the Guidelines, C-VI, 8.7). Copies of the documents are annexed to the communication and the numbering will be adhered to in the rest of the procedure:

D1: Wenger: Temporal scalability using P-pictures for low-latency applications, Multimedia Signal Processing, 1998 IEEE Second Workshop on Redondo Beach, CA, USA 7-9 Dec. 1998, Piscataway, NJ, USA,IEEE, US, pages 559-564, XP010318327

2 Claims 1, 13-15, 21-24 have been drafted as separate independent claims. Under Article 84 in combination with Rule 29(2) EPC an application may contain more than one independent claim in a particular category only if the subject matter claimed falls within one or more of the exceptional situations set out in paragraphs (a), (b) or (c) of Rule 29(2) EPC. The applicant is requested to amend the claims to comply with the above provisions.

3 The application does not meet the requirements of Article 84 EPC, because claim 1 is not clear.

Claim 1 lacks clarity as its subject-matter seems to be inconsistent with various embodiments of the description, e.g. that described in fig. 2 and in paras. 0035-0041.

Although claim 1 does not specify that the first and the second sub-sequences belong to the same layer, the term "sequence" implies consecutive frames and not different layers. Hence, according to claim 1, at least a picture in the second subsequence, which thus directly follows the first sub-sequence, is predicted from the first sub-sequence, i.e. "inter GOP prediction". However, according to fig. 2, there is no inter-GOP prediction but rather inter-layer prediction.

The applicant is requested to remove this inconsistency or explain why the subject-matter of claim 1 falls within the embodiment of fig. 2 and in particular explain the term "sub-sequence".

4 The present application does not meet the requirements of Article 52(1) EPC, because the subject-matter of claim 1 is not, as far as it has been understood, new in the sense of Article 54(1) and (2) EPC.

The document D1 discloses the following features of claim 1, as construed in light of the embodiment disclosed in fig. 2 and paras 0035-0041 of the description:

a method for coding video frames for the purpose of forming a scalable, compressed video sequence (title) comprising video frames coded according to at least a first and a second frame format (see fig. 2 and page 3, lines 15-17: The base layer comprises I and P-pictures), the video frames of the first frame format being independent of the other video frames (I), and the video frames of the second frame format being predicted from at least one of the other video frames (P), comprising the steps of forming into the video sequence at least part of a first sub- sequence (fig. 2: the first sub-sequence is the base layer), at least part of which has been formed by coding video frames of the at least first frame format (page 3, lines 15-17: the base layer includes an I-frame); and forming into the video sequence at least a second sub-sequence (fig. 2: the enhancement layer), at least part of which has been formed by coding at least video frames of the second frame format (fig. 2), and at least one video frame of the second sub- sequence has been predicted from at least one video frame of the first sub- sequence (fig. 2: the enhancement layer is predicted from the base layer); and forming into the video sequence information indicating which video frames belong to the second sub-sequence (page 4, lines 6-8: the reference picture

used is signalled in the picture header, thus the frame interdependencies are present in the bit stream).

The subject-matter of claim 1 is therefore not new (Article 54(1) and (2) EPC).

5 The above objections with respect to claim 1 directed to a method of coding apply *mutatis mutandis* to the corresponding independent claims 13-15 and 21-24.

6 The applicant is invited to file new claims which take account of the above objections. The applicant is also requested to explain how any independent claim differs from the method of D1.

7 It is suggested to defer any amendment to the description until an agreement as to an allowable set of claims has been reached.

—

Annika Wahrenberg
First examiner

EPA/EPO/OEB
D-80298 München
+49 89 2399-0
TX 523 656 epmu d
FAX +49 89 2399-4465

Europäisches Patentamt
Generaldirektion 2

European Patent Office
Directorate General 2

Office européen des brevets
Direction Générale 2

Telephone numbers: **Branch at The Hague**

Primary Examiner (substantive examination) +31 70 340-4890

Formalities Officer / Assistant (Formalities and other matters) +31 70 340-0

Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy,
Hermiankatu 12 B
33720 Tampere
FINLANDE

Application No.	Ref.	Date
03 700 321.7 - 1247	TP103660	16.06.2006
Applicant: Nokia Corporation		

Communication pursuant to Article 96(2) EPC

The examination of the above-identified application has revealed that it does not meet the requirements of the European Patent Convention for the reasons enclosed herewith. If the deficiencies indicated are not rectified the application may be refused pursuant to Article 97(1) EPC.

You are invited to file your observations and insofar as the deficiencies are such as to be rectifiable, to correct the indicated deficiencies within a period

of 4 months

from the notification of this communication, this period being computed in accordance with Rules 78(2) and 83(2) and (4) EPC.

One set of amendments to the description, claims and drawings is to be filed within the said period on separate sheets (Rule 36(1) EPC).

Failure to comply with this invitation in due time will result in the application being deemed to be withdrawn (Article 96(3) EPC).

Wahrenberg, A.
Primary Examiner
for the Examining Division

Enclosure(s): 3 page/s reasons (Form 2906)

Registered Letter
EPO Form 2001 07.02CSX

EPA/EPO/OEB
D-80298 München
+49 89 2399-0
TX 523 656 epmu d
FAX +49 89 2399-4465

Europäisches Patentamt | European Patent Office | Office européen des brevets

Generaldirektion 2 | Directorate General 2 | Direction Générale 2

COPY

Pursiainen, Timo Pekka
Tampereen Patenttitoimisto Oy,
Hermiankatu 12 B
33720 Tampere
FINLANDE

Datum/Date
08/04/05

Zeichen/Ref./Réf.	Anmeldung Nr./Application No./Demande n°./Patent Nr./Patent No./Brevet n°.
TP103660	03700321.7-2223/1479245

Anmelder/Applicant/Demandeur/Patentinhaber/Proprietor/Titulaire
Nokia Corporation

Communication of amended entries concerning the representative
(Rule 92 (1) (h) EPC)

As requested, for the above-mentioned European patent application /European patent the entries concerning the representative have been amended as follows:

Pursiainen, Timo Pekka, et al
Tampereen Patenttitoimisto Oy, Hermiankatu 12 B
33720 Tampere/FI

The amendment will be recorded in the Register of European Patents and published in the European Patent Bulletin (I.12/II.12).

Formalities Officer
Tel.No.: (089)2399

Europäisches Patentamt • European Patent Office • Office européen des brevets

PATENTIT • PATENTS
TAVARAMERKIT • TRADEMARKS
MALLIT • DESIGNS
HYÖDYLLISYYSMALLIT • UTILITY MODELS

TAMPEREEN PATENTTITOIMISTO

30 March 2005

VIA FACSIMILE

European Patent Office
Directorate General 2
D-80298 MÜNCHEN
Germany

M. 30.03.05 UST

CHANGE OF PROFESSIONAL REPRESENTATIVE

The professional representative has been changed in the applications and patents mentioned in the enclosed list. All future correspondence in these cases should be sent to

Mr. Timo Pursiainen

Tampereen Patenttitoimisto Oy
Hermiankatu 12 B
FI-33720 TAMPERE, Finland
Tel. +358-3-2886111
Fax. + 358-3-2886262
Email: tampat@tampereenpatenttitoimisto.fi

Please record the change and send us a confirmation as soon as possible.

Yours very truly,

Timo Pursiainen
Professional Representative

Application No.	Registration No.	Applicant	Our reference
01000602.1-2412		NOKIA CORPORATION	TP103653
03700321.7-2223		NOKIA CORPORATION	TP103660
02706807.1-2411		NOKIA CORPORATION	TP103671
01907602.5	1266650	NOKIA CORPORATION	TP103698
00912694.7		NOKIA NETWORKS OY	TP103714
02753110.2		NOKIA CORPORATION	TP103720
99952667.6		NOKIA NETWORKS OY	TP103729
99968279.2		NOKIA CORPORATION	TP103749
97901103.8		NOKIA TELECOMM.	TP103760
02102759.4-2414		NOKIA CORPORATION	TP103770
01907601.7	1269710	NOKIA CORPORATION	TP103778
99963607.9-2412		NOKIA CORPORATION	TP103790
01951730.9	1304001	NOKIA CORPORATION	TP103798

TAMPEREEN
PATENTTITOIMISTO

Hermiankatu 12 B, FIN-33720 Tampere
Tel +358 3 288 6111, Fax +358 3 288 6262

Receipt of Electronic Submission

We hereby acknowledge receipt of the following subsequently filed document(s):

Submission Number	42711	
Application Number	EP03700321.7	
Date of Receipt	9 March 2005	
Receiving Office	European Patent Office, The Hague	
Your Reference	2011658	
Applicant	Nokia Corporation	
Number of Applicants	1	
Documents Submitted	EPF1038.PDF ep-sfd-request.xml	PRIOTRAN-1.PDF package-data.xml
Submitted by	CN=S. Holmström 2091,O=Kolster Oy AB,C=FI	
Method of submission	Online	
Timestamp of Receipt	9 March 2005 10:46:20	
Official Digest of Submission	AB:D6:56:88:92:17:C2:10:BA:CB:A4:0F:34:0D:DC:6B:03:E9:71:F8	

/European Patent Office/

Europäisches Patentamt | European Patent office | Office européen des brevets

Sender:

Mr. Juha Kaukonen
Kolster Oy Ab
Iso Roobertinkatu 23|P.O. Box 148
FIN-00121 Helsinki
Finland

✉ D-80298 München
☎ (+49-89) 2399-0
Tx 523 656 epmu d
Fax (+49-89) 23 99-44 65

✉ P.B. 5818 Patentlaan 2
NL-2280 HV Rijswijk
☎ (+31-70) 340-2040
Tx 31 651 epo nl
Fax (+31-70) 340-3016

✉ D-10958 Berlin
☎ (+49-30) 25901-0
Fax (+49-30) 25901-840

LETTER ACCOMPANYING SUBSEQUENTLY FILED ITEMS

The document(s) listed below is (are) subsequently filed documents pertaining to the following application:

Application number	03700321.7
Applicant's or representative's reference	2011658

	Description of document	Original file name	Assigned file name
1	Translation of priority document	2011658PD.PDF	PRIOTRAN-1.PDF

Annotations

Signatures

Date 9 Mar 2005

Signed by: Subject: FI, Kolster Oy AB, S. Holmström 2091
Issuer: , European Patent Office, European Patent Office CA

Capacity: (Representative)

GROUPING OF IMAGE FRAMES IN VIDEO CODING

FIELD OF THE INVENTION

[0001] The invention relates to the grouping of multimedia files, particularly video files and particularly in connection with streaming.

BACKGROUND OF THE INVENTION

[0002] The term 'streaming' refers to simultaneous sending and playback of data, typically multimedia data, such as audio and video files, in which the recipient may begin data playback already before all the data to be transmitted has been received. Multimedia data streaming systems comprise a streaming server and terminal devices that the recipients use for setting up a data connection, typically via a telecommunications network, to the streaming server. From the streaming server the recipients retrieve either stored or real-time multimedia data, and the playback of the multimedia data can then begin, most advantageously almost in real-time with the transmission of the data, by means of a streaming application included in the terminal.

[0003] From the point of view of the streaming server, the streaming may be carried out either as normal streaming or as progressive downloading to the terminal. In normal streaming the transmission of the multimedia data and/or the data contents are controlled either by making sure that the bit rate of the transmission substantially corresponds to the playback rate of the terminal device, or, if the telecommunications network used in the transmission causes a bottleneck in data transfer, by making sure that the bit rate of the transmission substantially corresponds to the bandwidth available in the telecommunications network. In progressive downloading the transmission of the multimedia data and/or the data contents do not necessarily have to be interfered with at all, but the multimedia files are transmitted as such to the recipient, typically by using transfer protocol flow control. The terminals then receive, store and reproduce an exact copy of the data transmitted from the server, which copy can then be later reproduced again on the terminal without needing to start a streaming again via the telecommunications network. The multimedia files stored in the terminal are, however, typically very large and their transfer to the terminal is time-consuming, and they require a significant amount of memory capacity, which is why normal streaming is often preferred.

[0004] The video files in multimedia files comprise a great number of still image frames, which are displayed rapidly in succession (of typically 15

to 30 frames per s) to create an impression of a moving image. The image frames typically comprise a number of stationary background objects, determined by image information which remains substantially unchanged, and few moving objects, determined by image information that changes to some extent. The information comprised by consecutively displayed image frames is typically largely similar, i.e. successive image frames comprise a considerable amount of redundancy. The redundancy appearing in video files can be divided into spatial, temporal and spectral redundancy. Spatial redundancy refers to the mutual correlation of adjacent image pixels, temporal redundancy refers to the changes taking place in specific image objects in subsequent frames, and spectral redundancy to the correlation of different colour components within an image frame.

[0005] To reduce the amount of data in video files, the image data can be compressed into a smaller form by reducing the amount of redundant information in the image frames. In addition, while encoding, most of the currently used video encoders downgrade image quality in image frame sections that are less important in the video information. Further, many video coding methods allow redundancy in a bit stream coded from image data to be reduced by efficient, lossless coding of compression parameters known as VLC (Variable Length Coding).

[0006] In addition, many video coding methods make use of the above-described temporal redundancy of successive image frames. In that case a method known as motion-compensated temporal prediction is used, i.e. the contents of some (typically most) of the image frames in a video sequence are predicted from other frames in the sequence by tracking changes in specific objects or areas in successive image frames. A video sequence always comprises some compressed image frames the image information of which has not been determined using motion-compensated temporal prediction. Such frames are called INTRA-frames, or I-frames. Correspondingly, motion-compensated video sequence image frames predicted from previous image frames are called INTER-frames, or P-frames (Predicted). The image information of P-frames is determined using one I-frame and possibly one or more previously coded P-frames. If a frame is lost, frames dependent on it can no longer be correctly decoded.

[0007] An I-frame typically initiates a video sequence defined as a Group of Pictures (GOP), the P-frames of which can only be determined on the

basis of the I-frame and the previous P-frames of the GOP in question. The next I-frame begins a new group of pictures GOP, the image information comprised by which cannot thus be determined on the basis of the frames of the previous GOP. In other words, groups of pictures GOP are not temporally overlapping, and each group of picture can be decoded separately. In addition, many video compression methods employ bi-directionally predicted B-frames (Bi-directional), which are set between two anchor frames (I- and P-frames, or two P-frames) within a group of pictures GOP, the image information of a B-frame being predicted from both the previous anchor frame and the one succeeding the B-frame. B-frames therefore provide image information of higher quality than P-frames, but typically they are not used as anchor frames, and therefore their removal from the video sequence does not degrade the quality of subsequent images. However, nothing prevents B-frames from being used as anchor frames as well, only in that case they cannot be removed from the video sequence without deteriorating the quality of the frames dependent on them.

[0008] Each video frame can be divided into what are known as macroblocks that comprise the colour components (such as Y, U, V) of all pixels of a rectangular image area. More specifically, a macroblock consists of at least one block per colour component, the blocks each comprising colour values (such as Y, U or V) of one colour level in the image area concerned. The spatial resolution of the blocks may differ from that of the macroblocks, for example U- and V-components may be displayed using only half of the resolution of a Y-component. Macroblocks can be further grouped into slices, for example, which are groups of macroblocks that are typically selected in the scanning order of the image. Temporal prediction is typically carried out in video coding methods block- or macroblock-specifically, instead of image-frame-specifically.

[0009] To allow for flexible streaming of video files, many video coding systems employ scalable coding in which some elements or element groups of a video sequence can be removed without affecting the reconstruction of other parts of the video sequence. Scalability is typically implemented by grouping the image frames into a number of hierarchical layers. The image frames coded into the image frames of the base layer substantially comprise only the ones that are compulsory for the decoding of the video information at the receiving end. The base layer of each group of

pictures GOP thus comprises one I-frame and a necessary number of P-frames. One or more enhancement layers can be determined below the base layer, each one of the layers improving the quality of the video coding in comparison with an upper layer. The enhancement layers thus comprise P- or B-frames predicted on the basis of motion-compensation from one or more upper layer images. The frames are typically numbered according to an arithmetical series.

[0010] In streaming, transmission bit rate must be controllable either on the basis of the bandwidth to be used or the maximum decoding or bit rate value of the recipient. Bit rate can be controlled either at the streaming server or in some element of the telecommunications network, such as an Internet router or a base station of a mobile communications network. The simplest means for the streaming server to control the bit rate is to leave out B-frames having a high information content from the transmission. Further, the streaming server may determine the number of scalability layers to be transmitted in a video stream, and thus the number of the scalability layers can be changed always when a new group of pictures GOP begins. It is also possible to use different video sequence coding methods. Correspondingly, also other P-frames of the enhancement layers, in addition to the B-frames, can be removed from the bit stream in a telecommunications network element.

[0011] The above arrangement involves a number of problems. Many coding methods, such as the coding according to the ITU-T (International Telecommunications Union, Telecommunications Standardization Sector) standard H.263, are familiar with a procedure called reference picture selection. In reference picture selection at least a part of a P-image has been predicted from at least one other image than the one immediately preceding the P-image in the time domain. The selected reference image is signalled in a coded bit stream or in bit stream header fields image-, image-segment- (such as a slice or a group of macroblocks), macroblock-, or block-specifically. The reference picture selection can be generalized such that the prediction can also be made from images temporally succeeding the image to be coded. Further, the reference picture selection can be generalized to cover all temporally predicted frame types, including B-frames. Since it is possible to also select at least one image preceding an I-image that begins a group of pictures GOP as the reference image, a group of pictures employing reference picture selection cannot necessarily be decoded independently. In

addition, the adjusting of scalability or coding method in the streaming server or a network element becomes difficult, because the video sequence must be decoded, parsed and buffered for a long period of time to allow any dependencies between different image groups to be detected.

[0012] Another problem in the prior art coding methods is that there is no useful method of signalling differences of significance between INTER-frames. For example, when a plural number of P-frames are to be predicted successively, the first P-frames are usually the most significant ones for the reconstruction because there are more image frames that are dependent on the first P-frame than on subsequent P-frames. However, known coding methods fail to provide a simple method for signalling such differences in significance.

[0013] A further problem relates to the insertion of a video sequence in the middle of another video sequence, which has typically led to discontinuity in image numbering. The numbering of video sequence images is typically used for detecting the loss of image frames. However, if a separate video sequence, such as a commercial, is inserted into a video sequence, the separate video sequence is typically provided with separate image numbering, which is not in line with the ascending image numbering of the original video sequence. The receiving terminal may therefore interpret the deviating image numbering as a signal of lost image frames and start unnecessary actions to reconstruct the image frames suspected as lost or to request a re-transmission thereof. A similar problem is encountered when scaling a video sequence: for example, if a plural number of successive frames is removed, the receiving terminal may unnecessarily interpret these removals as protocol errors.

BRIEF DESCRIPTION OF THE INVENTION

[0014] It is therefore an object of the invention to provide a method and equipment implementing the method that allow the disadvantages caused by the above problems to be reduced. The objects of the invention are achieved by a method, video encoder, video decoder, streaming system element and computer programs that are chacterized by what is stated in the independent claims.

[0015] The preferred embodiments of the invention are disclosed in the dependent claims.

[0016] The invention is based on the idea of coding a scalable, compressed video sequence that comprises video frames coded on the basis of at least a first and a second frame format, the video frames conforming to the first frame format being independent of other video frames, i.e. they are typically I-frames, and the video frames according to the second frame format being frames predicted from at least one other video frame, for example P-frames. A video sequence comprises a first sub-sequence formed therein, at least part of the sub-sequence being formed by coding video frames (I-frames) of the at least first frame format (I-frames), and at least a second sub-sequence, at least part of which is formed by coding video frames of the at least second frame format (e.g. P-frames), at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence. In addition, identifying data of the video frames comprised by the at least second sub-sequence are determined into the video sequence in question.

[0017] Consequently, an essential aspect of the present invention is to determine the sub-sequences each sub-sequence is dependent on, i.e. a sub-sequence will comprise information of all sub-sequences that have been directly used for predicting the image frames comprised by the sub-sequence in question. This information is signalled in the bit stream of the video sequence, preferably separate from the actual image information, whereby the image data comprised by the video sequence can preferably be scaled, because independently decodable portions of the video sequence can be easily determined, and the portions can be removed without affecting the decoding of the rest of the image data.

[0018] According to a preferred embodiment of the invention, a scalable coding hierarchy is formed for the video sequences, according to which hierarchy a first scalability layer of the video sequence is coded in such a way that it comprises at least video frames of the first frame format, i.e. I-frames, and the lower scalability layers of the video sequence are coded such that they comprise at least video frames of the second frame format, i.e. P- and/or B-frames, grouped into sub-sequences in which at least one video frame is predicted either from a video frame of an upper scalability layer or from another video frame of the same sub-sequence. The number of the scalability layers is not restricted.

[0019] According to a preferred embodiment of the invention, a unique identifier combining the scalability layer number, sub-sequence identifier and image number is determined for each video frame. The identifier can be included into a header field of the video sequence, or into a header field according to the transfer protocol to be used for the video sequence transfer.

[0020] An advantage of the procedure of the invention is that it provides a flexible coding hierarchy that enables, on one hand, the scaling of the bit rate of a video sequence to be transmitted without video sequence decoding being required, and, on the other hand, independent decoding of each sub-sequence. This allows the streaming server, for example, to adjust the bit rate advantageously, without video sequence decoding, parsing and buffering, because the streaming server is capable of deducing the inter-dependencies of the different sub-sequences directly from the identifiers and their inter-dependencies. Further, the streaming server may, when necessary, carry out sub-sequence-specific adjusting of the scalability or the coding employed, because the inter-dependencies of the different frames are known. A further advantage is that the coding hierarchy and the image numbering of the invention allow a separate video sequence to the easily inserted into another video sequence.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] In the following, the invention will be described in connection with the preferred embodiments and with reference to the accompanying drawings, in which

Figure 1 illustrates a common multimedia data streaming system in which the scalable coding hierarchy of the invention can be applied;

Figure 2 illustrates a scalable coding hierarchy of a preferred embodiment of the invention;

Figures 3a and 3b illustrate embodiments of the invention for adjusting scalability;

Figures 4a, 4b and 4c illustrate embodiments of the invention for adjusting image numbering;

Figures 5a, 5b and 5c illustrate embodiments of the invention for using B-frames in a scalable coding hierarchy;

Figures 6a, 6b and 6c illustrate scalable coding hierarchies of preferred embodiments of the invention in connection with reference picture selection; and

Figure 7 illustrates an arrangement according to a preferred embodiment of the invention for coding scene transition.

DETAILED DESCRIPTION OF THE INVENTION

[0022] In the following, a general-purpose multimedia data streaming system is disclosed, the basic principles of which can be applied in connection with any telecommunications system. Although the invention is described here with a particular reference to a streaming system, in which multimedia data is transmitted most preferably through a telecommunications network employing a packet-switched data protocol, such as an IP network, the invention can equally well be implemented in circuit-switched networks, such as fixed telephone networks PSTN/ISDN (Public Switched Telephone Network/Integrated Services Digital Network) or in mobile communications networks PLMN (Public Land Mobile Network). Further, the invention can be applied in the streaming of multimedia files in the form of both normal streaming and progressive downloading, and for implementing video calls, for example.

[0023] It is also to be noted that although the invention is described here with a particular reference to streaming systems and the invention can also be advantageously applied in them, the invention is not restricted to streaming systems alone, but can be applied in any video reproduction system, irrespective of how a video file that is to be decoded is downloaded and where it is downloaded from. The invention can therefore be applied for example in the playback of a video file to be downloaded from a DVD disc or from some other computer memory carrier, for example in connection with varying processing capacity available for video playback. In particular, the invention can be applied to different video codings of low bit rate that are typically used in telecommunications systems subject to bandwidth restrictions. One example is the system defined in the ITU-T standard H.263 and the one that is being defined in H.26L (possibly later to become H.264). In connection with these, the invention can be applied to mobile stations, for example, in which case the video playback can be made to adjust both to changing transfer capacity or channel quality and to the processor power currently available, when the

mobile station is used also for executing other applications than video playback.

[0024] It is further to be noted that, for the sake of clarity, the invention will be described below by giving an account of image frame coding and temporal predicting on image frame level. However, in practice coding and temporal predicting typically take place on block or macroblock level, as described above.

[0025] With reference to Figure 1, a typical multimedia streaming system will be described, which is a preferred system for applying the procedure of the invention.

[0026] A multimedia data streaming system typically comprises one or more multimedia sources 100, such as a video camera and a microphone, or video image or computer graphic files stored in a memory carrier. Raw data obtained from the different multimedia sources 100 is combined into a multimedia file in an encoder 102, which can also be referred to as an editing unit. The raw data arriving from the one or more multimedia sources 100 is first captured using capturing means 104 included in the encoder 102, which capturing means can be typically implemented as different interface cards, driver software, or application software controlling the function of a card. For example, video data may be captured using a video capture card and the associated software. The output of the capturing means 104 is typically either an uncompressed or slightly compressed data flow, for example uncompressed video frames of the YUV 4:2:0 format or motion-JPEG image format, when a video capture card is concerned.

[0027] An editor 106 links different media flows together to synchronize video and audio flows to be reproduced simultaneously as desired. The editor 106 may also edit each media flow, such as a video flow, by halving the frame rate or by reducing spatial resolution, for example. The separate, although synchronized, media flows are compressed in a compressor 108, where each media flow is separately compressed using a compressor suitable for the media flow. For example, video frames of the YUV 4:2:0 format may be compressed using the low bit rate video coding according to the ITU-T recommendation H.263 or H.26L. The separate, synchronized and compressed media flows are typically interleaved in a multiplexer 110, the output obtained from the encoder 102 being a single, uniform bit flow that comprises data of a plural number of media flows and that may be referred to

as a multimedia file. It is to be noted that the forming of a multimedia file does not necessarily require the multiplexing of a plural number of media flows into a single file, but the streaming server may interleave the media flows just before transmitting them.

[0028] The multimedia files are transferred to a streaming server 112, which is thus capable of carrying out the streaming either as a real-time streaming or in the form of progressive downloading. In progressive downloading the multimedia files are first stored in the memory of the server 112 from where they may be retrieved for transmission as need arises. In real-time streaming the editor 102 transmits a continuous media flow of multimedia files to the streaming server 112, and the server 112 forwards the flow directly to a client 114. The streaming server 112 carries out traffic shaping of the multimedia data as regards the bandwidth available or the maximum decoding and playback rate of the client 114, the streaming server being able to adjust the bit rate of the media flow for example by leaving out B-frames from the transmission or by adjusting the number of the scalability layers. Further, the streaming server 112 may modify the header fields of a multiplexed media flow to reduce their size and encapsulate the multimedia data into data packets that are suitable for transmission in the telecommunications network employed. The client 114 may typically adjust, at least to some extent, the operation of the server 112 by using a suitable control protocol. The client 114 is capable of controlling the server 112 at least in such a way that a desired multimedia file can be selected for transmission to the client, in addition to which the client is typically capable of stopping and interrupting the transmission of a multimedia file.

[0029] When the client 114 is receiving a multimedia file, the file is first supplied to a demultiplexer 116, which separates the media flows comprised by the multimedia file. The separate, compressed media flows are then supplied to a decompressor 118 where each separate media flow is decompressed by a decompressor suitable for each particular media flow. The decompressed and reconstructed media flows are supplied to a playback unit 120 where the media flows are rendered at a correct pace according to their synchronization data and supplied to presentation means 124 for reproduction. The actual presentation means 124 may comprise for example a computer or mobile station display, and loudspeaker means. The client 114 also typically comprises a control unit 122 that the end user can typically control through a

user interface and that controls both the operation of the server, through the above-described control protocol, and the operation of the playback unit 120, on the basis of the instructions given by the end user.

[0030] It is to be noted that the transfer of multimedia files from the streaming server 112 to the client 114 takes place through a telecommunications network, the transfer path typically comprising a plural number of telecommunications network elements. It is therefore possible that there is at least some network element that can carry out traffic shaping of multimedia data with regard to the available bandwidth or the maximum decoding and playback rate of the client 114 at least partly in the same way as described above in connection with the streaming server.

[0031] Scalable coding will be described below with reference to a preferred embodiment of the invention and an example illustrated in Figure 2. Figure 2 shows part of a compressed video sequence having a first frame 200, which is an INTRA frame, or I-frame, and thus an independently determined video frame the image information of which is determined without using motion-compensated temporal prediction. The I-frame 200 is placed on a first scalability layer, which may be referred to as an INTRA layer. Each scalability layer is assigned a unique identifier, such as the layer number. The INTRA layer may therefore be given the number 0, for example, or some other alphanumeric identifier, for example a letter, or a combination of a letter and a number.

[0032] Correspondingly, sub-sequences consisting of groups of one or more video frames are determined for each scalability layer, at least one of the images in a group (typically the first or the last one) being temporally predicted at least from a video frame of another sub-sequence of typically either a higher or the same scalability layer, the rest of the video frames being temporally predicted either from only the video frames of the same sub-sequence, or possibly also from one or more video frames of said second sub-sequence. A sub-sequence may be decoded independently, irrespective of other sub-sequences, except said second sub-sequence. The sub-sequences of each scalability layer are assigned a unique identifier using for example consecutive numbering starting with number 0 given for the first sub-sequence of a scalability layer. Since the I-frame 200 is determined independently and can also be decoded independently upon reception, irrespective of other image frames, it also forms in a way a separate sub-sequence.

[0033] An essential aspect of the present invention is therefore to determine each sub-sequence in terms of those sub-sequences the sub-sequence is dependent on. In other words, a sub-sequence comprises information about all the sub-sequences that have been directly used for predicting the image frames of the sub-sequence in question. This information is signalled in a video sequence bit stream, preferably separate from the actual image information, and therefore the image data of the video sequence can be preferably adjusted because it is easy to determine video sequence portions that are to be independently decoded and can be removed without affecting the decoding of the rest of the image data.

[0034] Next, within each sub-sequence, the video frames of the sub-sequence are given image numbers, using for example consecutive numbering that starts with the number 0 given to the first video frame of the sub-sequence. Since the I-frame 200 also forms a separate sub-sequence, its image number is 0. In Figure 2, the I-frame 200 shows the type (I), sub-sequence identifier and image number (0.0) of the frame.

[0035] Figure 2 further shows a next I-frame 202 of the INTRA layer, the frame thus being also an independently determined video frame that has been determined without using motion-compensated temporal prediction. The temporal transmission frequency of I-frames depends on many factors relating to video coding, image information contents and the bandwidth to be used, and, depending on the application or application environment, the I-frames are transmitted in a video sequence at intervals of 0.5 to 10 seconds, for example. Since the I-frame 202 can be independently decoded, it also forms a separate sub-sequence. Since this is the second sub-sequence in the INTRA-layer, the consecutive numbering of the sub-sequence identifier of the I-frame 202 is 1. Further, since the I-frame 202 also forms a separate sub-sequence, i.e. it is the only video frame in the sub-sequence, its image number is 0. The I-frame 202 can thus be designated with identifier (I.1.0.) Correspondingly, the identifier of the next I-frame on the INTRA layer is (I.2.0.), etc. As a result, only independently determined I-frames in which the image information is not determined using motion-compensated temporal prediction are coded into the first scalability layer, i.e. the INTRA layer. The sub-sequences can also be determined using other kind of numbering or other identifiers, provided that the sub-sequences can be distinguished from one another.

[0036] The next scalability layer, which has a layer number 1, for example, and which may be referred to as the base layer, comprises coded, motion-compensated INTER or P-frames typically predicted only from previous image frames, i.e. in this case from the I-frames of an upper INTRA layer. The image information of the first P-frame 204 of the base layer shown in Figure 2 is determined using the I-frame 200 of the INTRA layer. A P-frame 204 begins the first sub-sequence of the base layer and therefore the sub-sequence identifier of the P-frame 204 is 0. Further, since the P-frame 204 is the first image frame of the first sub-sequence of the base layer, the image number of the P-frame 204 is 0. The P-frame 204 can thus be identified with (P.0.0).

[0037] The temporally succeeding P-frame 206 of the base layer is predicted from the previous P-frame 204. The P-frames 204 and 206 thus belong to the same sub-sequence, whereby the P-frame 206 also receives the sub-sequence identifier 0. Since the P-frame 206 is the second image frame in the sub-sequence 0, the image number of the P-frame 206 is 1, and the P-frame 206 can be identified with (P.0.1).

[0038] The scalability layer succeeding the base layer and having the layer number 2, is called the first enhancement layer. This layer comprises coded, motion-compensated P-frames predicted only from previous image frames, in this case either from I-frames of the INTRA layer or P-frames of the base layer. Figure 2 shows a first image frame 208 and a second image frame 210 of the first enhancement layer, which are both predicted only from the first image frame 200 of the INTRA layer. The P-frame 208 begins the first sub-sequence of the first enhancement layer, the sub-sequence identifier of the P-frame 208 thus being 0. Further, since the P-frame 208 is the first and only image frame in said sub-sequence, the P-frame 208 receives the image number 0. The P-frame 208 can thus be identified with (P.0.0.).

[0039] Since the second image frame 210 is also predicted only from the first image frame 200 of the INTRA layer, the P-frame 210 begins the second sub-sequence of the first enhancement layer and the sub-sequence identifier of the P-frame 210 is therefore 1. Since the P-frame 210 is the first image frame in the sub-sequence, the image number of the P-frame 210 is 0. The P-frame can thus be identified with (P.1.0.). The temporally succeeding P-frame 212 of the first enhancement layer is predicted from the previous P-frame 210. The P-frames 210 and 212 thus belong to the same sub-sequence, and therefore the P-frame also receives the sub-sequence identifier 1. The P-

frame 212 is the second image frame in the sub-sequence 1, and therefore the P-frame 212 receives the image number 1 and can be identified with (P.1.1).

[0040] The temporally fourth image frame 214 of the first enhancement layer is predicted from the first image frame 204 of the base layer. The P-frame 214 thus begins a third sub-sequence of the first enhancement layer and therefore the P-frame 214 receives the sub-sequence identifier 2. Further, since the P-frame 214 is the first and only image frame in the sub-sequence, the image number of the P-frame 214 is 0. The P-frame 208 can therefore be identified with (P.2.0).

[0041] Also the temporally fifth image frame 216 of the first enhancement layer is predicted only from the first image frame 204 of the base layer, and therefore the P-frame 216 begins the fourth sub-sequence of the first enhancement layer, the sub-sequence identifier of the P-frame 216 being 3. In addition, since the P-frame 216 is the first one in the sub-sequence in question, the image number of the P-frame 216 is 0. The P-frame 216 can therefore be identified with (P.3.0). The temporally following P-frame 218 of the first enhancement layer is predicted from the previous P-frame 216. The P-frames 216 and 218 thus belong to the same sub-sequence and thus the sub-sequence identifier of the P-frame 218 is also 3. Since the P-frame 218 is the second image frame in sub-sequence 3, the image number of the P-frame 218 is 1 and the identifier of the P-frame 218 is (P.3.1).

[0042] For simplicity and clarity of illustration the above disclosure only relates to I- and P-frames. However, a person skilled in the art will find it apparent that the scalable video coding of the invention can also be implemented using other known image frame types, such as the above described B-frames and at least SI-frames, SP-frames and MH-frames. SI-frames correspond to I-frames, but together with an SP-frame they allow an identical image to be reconstructed. An SP-frame, in turn, is a P-frame subjected to a particular coding that allows an identical image to be reconstructed together with an SI-frame or another SP-frame. SP-frames are typically placed into video sequence points, where an access point or a scanning point is desired, or where the changing of the coding parameters of the video stream should be possible. The frames can also be used for error correction and for increasing error tolerance. SP-frames are otherwise similar to ordinary P-frames predicted from previous frames, except that they are defined so that they can be replaced by another video frame of the SP- or SI-

type, the result of the decoding of the new frame being identical with the decoding result of the original SP-frame that was in the video stream. In other words, a new SP-frame that is used for replacing the SP-frame that was in the video stream is predicted from another sequence or video stream and yet the reconstructed frame has identical contents. SP-frames are described for example in the Applicant's earlier application PCT/FI02/00004.

[0043] Similarly as B-frames, macroblocks of MH (Multi Hypothesis) frames, based on motion-compensated prediction, are predicted from two other frames, which are not, however, necessarily located next to an MH-frame. More precisely, the predicted macroblocks are computed as an average of two macroblocks of two other frames. Instead of two frames, MH-frame macroblocks can naturally be also predicted from one other frame. Reference images may change according to macroblock, in other words, all macroblocks in one and the same image are not necessarily predicted using the same frames.

[0044] A sub-sequence thus covers a specific period of time in a video sequence. The sub-sequences of the same layer or of different layers may be partly or entirely overlapping. If there are temporally overlapping image frames on the same layer, the frames are interpreted as alternative presentations of the same image content and therefore any mode of image presentation can be used. On the other hand, if there are temporally overlapping image frames on different layers, they form different presentations of the same image content, and therefore presentations differ in image quality, i.e. the quality of image is better on a lower layer.

[0045] The above disclosure referring to Figure 2 illustrates a scalable coding arrangement and a hierarchical structure and numbering of image frames according to a preferred embodiment of the invention. In this embodiment the INTRA-layer only comprises I-frames and the base layer can only be decoded using the information received from the INTRA-layer. Correspondingly, the decoding of the first enhancement layer typically requires information from both the base layer and the INTRA-layer.

[0046] The number of the scalability layers is not restricted to three, as above, but any number of enhancement layers that is considered necessary for producing sufficient scalability may be used. Consequently, the layer number of the second enhancement layer is four, that of the third enhancement layer is five, etc. Since some of the image frames in the above

example are given the same identifier (e.g. the identifier of both image frames 204 and 208 is (P.0.0)), by including the layer number in the identifier each image frame can be uniquely identified and, at the same time, the dependencies of each image frame on other image frames is preferably determined. Each image frame is thus uniquely identified, the identifier of image frame 204, for example, being (P.1.0.0), or simply (1.0.0) and, correspondingly, that of image frame 208 being (P.2.0.0), or (2.0.0).

[0047] According to a preferred embodiment of the invention, the number of a reference image frame is determined according to a specific, predetermined alpha-numeric series, for example as an integer between 0 and 255. When the parameter value achieves the maximum value N (e.g. 255) in the series concerned, the determining of the parameter value starts from the beginning, i.e. from the minimum value of the series (e.g. 0). An image frame is thus uniquely identified within a specific sub-sequence up to the point where the same image number is used again. The sub-sequence identifier can also be determined according to a specific, predetermined arithmetic series. When the value of the sub-sequence identifier achieves the maximum value N of the series, the determining of the identifier starts again from the beginning of the arithmetic series. However, a sub-sequence cannot be assigned an identifier that is still in use (within the same layer). The series in use may also be determined in another way than arithmetically. One alternative is to assign random sub-sequence identifiers, taking into account that an assigned identifier is not used again.

[0048] A problem in the numbering of image frames arises when browsing of a video file is to be started in the middle of a video sequence. Such situations occur for example when the user wishes to browse a locally stored video file backward or forward or to browse a streaming file at a particular point; when the user initiates the playback of a streaming file from a random point; or when a video file that is to be reproduced is detected to contain an error that interrupts the playback or requires the playback to be resumed from a point following the error. When the browsing of a video file is resumed from a random point after previous browsing, discontinuity typically occurs in the image numbering and the decoder typically interprets this as unintentional loss of image frames and unnecessarily tries to reconstruct the image frames suspected as lost.

[0049] According to a preferred embodiment of the invention, this can be avoided in the decoder by defining an initiation image in an independently decodable Group of Pictures GOP that is activated at a random point of the video file and by setting the number of the initiation image at zero. This independently decodable image group can thus be a sub-sequence of an INTRA-layer, for example, in which case an I-frame is used as the initiation image, or, if scaling originating from the base layer is employed, the independently decodable image group is a sub-sequence of the base layer, in which case the first image frame of the sub-sequence, typically an I-frame, is used as the initiation image. Consequently, when activated at a random point, the decoder preferably sets the identifier of the first image frame, preferably an I-frame, of the independently decodable sub-sequence at zero. Since the sub-sequence to be decoded may also comprise other image frames whose identifier is zero (for example when the above described alpha-numeric series starts from the beginning), the beginning of the sub-sequence, i.e. its first image frame, can be indicated to the decoder for example by a separate flag added to the header field of a slice of the image frame. This allows the decoder to interpret the image numbers correctly and to find the correct image frame that initiates the sub-sequence among the video sequence image frames.

[0050] The above numbering system provides only one example of how the unique image frame identification of the invention can be carried out so that interdependencies between the image frames are indicated at the same time. However, video coding methods in which the method of the invention can be applied, such as video coding methods according to the ITU-T standards H.263 and H.26L, employ code tables, which in turn use variable length codes. When variable length codes are used for coding layer numbers, for example, a lower code word index, i.e. a smaller layer number, signifies a shorter code word. In practice the scalable coding of the invention will be used in most cases in such a way that the base layer will consist significantly more image frames than the INTRA-layer. This justifies the use of a lower index, i.e. a smaller layer number, on the base layer than on the INTRA-layer, because the amount of coded video data is thereby advantageously reduced. Consequently, the INTRA-layer is preferably assigned layer number 1 and the base layer is given layer number 0. Alternatively, the code can be formed by using fewer bits for coding the base layer number than the INTRA-layer

number, in which case the actual layer number value is not relevant in view of the length of the code created.

[0051] Further, according to a second preferred embodiment of the invention, when the number of the scalability layers is to be kept low, the first scalability layer in particular can be coded to comprise both the INTRA-layer and the base layer. From the point of view of coding hierarchy, the simplest way to conceive this is to leave out the INTRA-layer altogether and to provide the base layer with coded frames consisting of both independently defined I-frames, the image information of which has not been determined using motion-compensated temporal prediction, and image frames predicted from previous frames, which image frames in this case are motion-compensated P-frames predicted from the I-frames of the same layer. The layer number value 0 can thus still be used for the base layer and, if enhancement layers are coded into the video sequence, the first enhancement layer is assigned layer number 1. This is illustrated in the following, with reference to Figures 3a and 3b.

[0052] Figure 3a shows a non-scalable video sequence structure, in which all image frames are placed on the same scalability layer, i.e. the base layer. The video sequence comprises a first image frame 300 which is an I-frame (I.0.0) and which thus initiates a first sub-sequence. The image frame 300 is used for predicting a second image frame 302 of the sub-sequence, i.e. a P-frame (P.0.1), which is then used for predicting a third image frame 304 of the sub-sequence, i.e. a P-frame (P.0.2), which is in turn used for predicting the next image frame 306, i.e. a P-frame (P.0.3). The video sequence is then provided with an I-frame (I.1.0) coded therein, i.e. an I-frame 308, which thus initiates a second sub-sequence in the video sequence. This kind of non-scalable coding can be used for example when the application employed does not allow scalable coding to be used, or there is no need for it. In a circuit-switched videophone application, for example, channel bandwidth remains constant and the video sequence is coded in real-time and therefore there is typically no need for scalable coding.

[0053] Figure 3b, in turn, illustrates an example of how scalability can be added, when necessary, to a combined INTRA- and base layer. Here, too, the video sequence base layer comprises a first image frame 310 which is again an I-frame (I.0.0) and which initiates a first sub-sequence of the base layer. The image frame 310 is used for predicting a second image frame 312 of the sub-sequence, i.e. a P-frame (P.0.1), which is then used for predicting a

third image frame 314 of the sub-sequence, i.e. a P-frame (P.0.2). The first enhancement layer, however, is also coded into this video sequence and it comprises a first sub-sequence, the first and only image frame 316 of which is a P-frame (P.0.0), which is predicted from the first image frame 310 of the base layer. The first image frame 318 of a second sub-sequence of the enhancement layer is, in turn, predicted from the second image frame 312 of the base layer, and therefore the identifier of this P-frame is (P.1.0). The next image frame 320 of the enhancement layer is again predicted from the previous image frame 318 of the same layer and, therefore, it belongs to the same sub-sequence, its identifier thus being (P.1.1).

[0054] In this embodiment of the invention the sub-sequences of the base layer can be decoded independently, although a base layer sub-sequence may be dependent on another base layer sub-sequence. The decoding of the base layer sub-sequences requires information from the base layer and/or from the second sub-sequence of the first enhancement layer, the decoding of the sub-sequences of the second enhancement layer requires information from the first enhancement layer and/or from the second sub-sequence of the second enhancement layer, etc. According to an embodiment, I-frames are not restricted to the base layer alone, but lower enhancement layers may also comprise I-frames.

[0055] The basic idea behind the above embodiments is that a sub-sequence comprises information about all the sub-sequences it is dependent on, i.e. about all sub-sequences that have been used for predicting at least one of the image frames of the sub-sequence in question. However, according to an embodiment it is also possible that a sub-sequence comprises information about all sub-sequences that are dependent on the sub-sequence in question, in other words, about all the sub-sequences in which at least one image frame has been predicted using at least one image frame of the sub-sequence in question. Since in the latter case the dependencies are typically determined temporally forward, image frame buffers can be advantageously utilized in the coding in a manner to be described later.

[0056] In all the above embodiments the numbering of the image frames is sub-sequence-specific, i.e. a new sub-sequence always starts the numbering from the beginning. The identification of an individual image frame thus requires the layer number, sub-sequence identifier, and image frame number to be determined. According to a preferred embodiment of the

invention, the image frames can be independently numbered using consecutive numbering in which successive reference image frames in the coding order are indicated with numbers incremented by one. As regards layer numbers and sub-sequence identifiers, the above-described numbering procedure can be used. This allows each image frame to be uniquely identified, when necessary, without using the layer number and sub-sequence identifier.

[0057] This is illustrated with the example shown in Figure 4a in which the base layer comprises a temporally first I-frame 400 (I.0.0). This frame is used for predicting a first image frame 402 of the first enhancement layer, i.e. (P.0.1), which is then used for predicting a second image frame 404 (P.0.2) belonging to the same sub-sequence (with sub-sequence identifier 0), which is used for predicting a third image frame 406 (P.0.3) of the same sub-sequence, which is used for predicting a fourth image frame 408 (P.0.4) and, finally, the fourth frame for predicting a fifth image frame 410 (P.0.5). The temporally next video sequence image frame 412 is located on the base layer, where it is in the same sub-sequence as the I-frame 400, although temporally it is only the seventh coded image frame. Therefore its identifier is (P.0.6). The seventh frame is then used for predicting a first image frame 414 (P.1.7) of the second sub-sequence of the first enhancement layer, which is then used for predicting a second image frame 416 (P.1.8) belonging to the same sub-sequence (with sub-sequence identifier 1), which is in turn used for predicting a third image frame 418 (P.1.9), the third for predicting a fourth image frame 420 (P.1.10) and, finally, the fourth for predicting a fifth image frame 422 (P.1.11) of the same sub-sequence. Again, the temporally next video sequence image frame 424 is located on the base layer, where it is in the same sub-sequence as the I-frame 400 and the P-frame 412, although temporally it is only the thirteenth coded image frame and therefore its identifier is (P.0.12). For clarity of illustration, the above description of the embodiment does not comprise layer identifiers, but it is apparent that in order to implement scalability, also the layer identifier must be signalled together with the video sequence, typically as part of the image frame identifiers.

[0058] Figures 4b and 4c show alternative embodiments for grouping the image frames of the video sequence shown in Figure 4a. The image frames in Figure 4b are numbered according to sub-sequence, i.e. a new sub-sequence always starts the numbering from the beginning (from

zero). Figure 4c, in turn, employs image frame numbering which corresponds otherwise to that used in Figure 4a, except that the P-frames of the base layer are replaced by SP-frame pairs to allow for identical reconstruction of image information.

[0059] As stated above, the procedure of the invention can also be implemented using B-frames. One example of this is illustrated in Figures 5a, 5b and 5c. Figure 5a shows a video sequence in the time domain, the sequence comprising P-frames P1, P4 and P7, with B-frames placed between them, the interdependencies of the B-frames with regard to temporal predicting being shown with arrows. Figure 5b shows a preferred grouping of video sequence image frames in which the interdependencies shown in Figure 5a are indicated. Figure 5b illustrates sub-sequence-specific image frame numbering in which a new sub-sequence always starts the numbering of the image frames from the beginning (from zero). Figure 5c, in turn, illustrates image frame numbering, which is otherwise consecutive except that a P-frame following a B-frame always receives the same image number as the B-frame.

[0060] The above examples illustrate different alternatives of how scalability of video sequence coding can be adjusted by using the method of the invention. From the point of view of the terminal device reproducing the video sequence, the more scalability layers are available, or the more scalability layers it is capable of decoding, the better the image quality. In other words, increase in the amount of image information and in the bit rate used for transferring the information improves the temporal or spatial resolution or the spatial quality of the image data. Correspondingly, a higher number of scalability layers also sets considerably higher demands on the processing capacity of the terminal device performing decoding.

[0061] In addition, the above examples illustrate the advantage gained by using sub-sequences. With image frame identifiers, the dependencies of each image frame from other image frames in the sub-sequence are indicated in an unambiguous manner. A sub-sequence thus forms an independent whole that can be left out of a video sequence, when necessary, without affecting the decoding of subsequent image frames of the video sequence. In that case, only the image frames of the sub-sequence in question and of those sub-sequences on the same and/or lower scalability layers dependent on it are not decoded.

[0062] The image frame identifier data transmitted together with the video sequence are preferably included in the video sequence header fields or in the header fields of the transfer protocol to be used for transferring the video sequence. In other words, the identifier data of predicted image frames are not included in the image data of the coded video sequence, but always into the header fields, whereby the dependencies of the image frames can be detected without decoding the images of the actual video sequence. The identifier data of the image frames can be stored for example in the buffer memory of the streaming server as the video sequence is being coded for transmission. In addition, the sub-sequences can be independently decoded on each scalability layer, because the image frames of a sub-sequence are not dependent on other sub-sequences of the same scalability layer.

[0063] According to an embodiment of the invention, the image frames comprised by a sub-sequence may thus depend also on other sub-sequences of the same scalability layer. This dependency must then be signalled for example to the streaming server carrying out traffic shaping, because interdependent sub-sequences located on the same layer cannot be separately removed from a video sequence to be transmitted. A preferred way to carry out the signalling is to include it in the image frame identifiers to be transmitted, for example by listing the layer-sub-sequence pairs the sub-sequence in question depends on. This also provides a preferred way of indicating dependency from another sub-sequence of the same scalability layer.

[0064] The above examples illustrate a situation where image frames are temporally predicted from previous image frames. In some coding methods, however, the reference picture selection has been further extended to also include the predicting of the image information of image frames from temporally succeeding image frames. Reference picture selection offers most diversified means for creating different temporally scalable image frame structures and allows the error sensitivity of the video sequence to be reduced. One of the coding techniques based on reference picture selection is INTRA-frame postponement. The INTRA-frame is not placed into its temporally "correct" position in the video sequence, but its position is temporally postponed. The video sequence image frames that are between the "correct" position of the INTRA-frame and its actual position are predicted temporally backward from the INTRA-frame in question. This naturally requires that image

frames that are not coded be buffered for a sufficiently long period of time so that all image frames that are to be displayed can be coded and arranged into their order of presentation. INTRA-frame transfer and the associated determining of sub-sequences in accordance with the invention are illustrated in the following with reference to Figure 6.

[0065] Figure 6a shows a video sequence part in which the INTRA-frame comprises a single I-frame 600, which is temporally transferred to the position shown in Figure 6, although the "correct" position of the I-frame in the video sequence would have been at the first image frame. The video sequence image frames between the "correct" position and the real position 600 are thus temporally predicted backward from the I-frame 600. This is illustrated by a sub-sequence coded into the first enhancement layer and having a first temporally backward predicted image frame 602, which is a P-frame (P.0.0). This P-frame is used for predicting a temporally previous image frame 604, i.e. a P-frame (P.0.1), which is used in turn for predicting an image frame 606, i.e. a P-frame (P.0.2), and, finally, an image frame 608, i.e. a P-frame (P.0.3), which is at the temporal position that would have been the temporally "correct" position of the I-frame 600 in the video sequence. Correspondingly, the I-frame 600 on the base layer is used for a temporally forward prediction of a sub-sequence comprising four P-frames 610, 612, 614 and 616, i.e. P-frames (P.0.0), (P.0.1), (P.0.2) and (P.0.3).

[0066] The fact that in this example backward predicted image frames are placed on a lower layer than forward predicted image layers indicates that for purposes of illustration, backward predicted image frames are in this coding example considered subjectively less valuable than forward predicted image frames. Naturally the sub-sequences could both be placed on the same layer, in which case they would be considered equal, or a backward predicted sub-sequence could be on the upper layer, in which case it would be considered subjectively more valuable.

[0067] Figures 6b and 6c show some alternatives for coding a video sequence according to Figure 6a. In Figure 6b both forward and backward predicted sub-sequences are placed on the base layer, the I-frame being only located on the INTRA-layer. The forward predicted sub-sequence on this layer is thus the second sub-sequence and its sub-sequence identifier is 1. In Figure 6c, in turn, an I-frame and a forward predicted sub-sequence based on it are

located on the base layer, while a backward predicted sub-sequence is located on the first enhancement layer.

[0068] Moreover, according to a preferred embodiment of the invention, the above-described scalability can be utilized for coding what is known as a scene transition into a video sequence. Video material, such as news reports, music videos and movie trailers, often comprise rapid cuts between separate image material scenes. Sometimes the cuts are abrupt, but often a procedure known as scene transition is used in which transfer from one scene to another takes place by dimming, wiping, mosaic dissolving or scrolling the image frames of a previous scene, and, correspondingly, by presenting those of a later scene. From the point of view of coding efficiency, the video coding of a scene transition is often most problematic, because the image frames appearing during the scene transition comprise information on the image frames of both the terminating and the initiating scene.

[0069] A typical scene transition, fading, is carried out by gradually reducing the intensity or luminance of the image frames of a first scene to zero, while gradually increasing the intensity of the image frames of a second scene to its maximum value. This scene transition is referred to as cross-faded scene transition.

[0070] Generally speaking, a computer-made image can be thought of as consisting of layers, or image objects. Each image object can be defined with reference to at least three information types: the structure of the image object, its shape and transparency, and the layering order (depth) in relation to the background of the image and to other image objects. Shape and transparency are often determined using what is known as an alpha plane, which measures opacity and the value of which is usually determined separately for each image object, possibly excluding the background, which is usually determined as non-transparent. The alpha plane value of a non-transparent image object, such as the background, can thus be set at 1.0, whereas the alpha plane value of a fully transparent image object is 0.0. The values in between define the intensity of the visibility of a specific image object in a picture in proportion to the background and to other, at least partly overlapping image objects that have a higher depth value than the image object in question.

[0071] The superimposition of image objects in layers according to their shape, transparency and depth position is referred to as scene

composition. In practice the procedure is based on the use of weighted averages. First, the image object that is closest to the background, i.e. deepest according to its depth position, is placed onto the background and a combined image is formed of the two. The pixel values of the combined image are formed as an average weighted by the alpha plane values of the background image and the image object in question. The alpha plane value of the combined image is then set at 1.0, after which it serves as a background image for the next image object. The process continues until all image objects are attached to the image.

[0072] In the following, a procedure according to a preferred embodiment of the invention will be described in which video sequence scalability layers are combined with the above described image objects of image frames and their information types to provide a scene transition with scalable video coding that also has good compression efficiency.

[0073] This embodiment of the invention is illustrated in the following by way of example and in a simplified manner by using cross-faded scene transition, on one hand, and abrupt scene transition, on the other hand, as examples. The image frames to be displayed during a scene transition are typically formed of two superimposed image frames, a first image frame comprising a first image scene and a second image frame a second scene. One of the image frames serves as the background image, whereas the other, which is referred to as a foreground image, is placed on top of the background image. The opacity of the background image, i.e. its non-transparency value, is constant, in other words, its pixel-specific alpha plane values are not adjusted.

[0074] In this embodiment of the invention, the background and foreground images are both defined according to scalability layer. This is illustrated in Figure 7, which shows an example of how image frames of two different scenes can be placed on scalability layers during a scene transition of the invention. Figure 7 shows a first image frame 700 of a first (terminating) scene positioned on the base layer. The image frame 700 may be either an I-frame containing image information that has not been determined using motion-compensated temporal predicting, or it may be a P-frame that is a motion-compensated image frame predicted from previous image frames. The coding of a second (initiating) scene starts during the temporally following image frame and, according to the invention, the image frames of the scene are also placed on the base layer. Remaining image frames 702, 704 of the

second (terminating) scene are then placed on the first enhancement layer. These image frames are typically P-frames.

[0075] In this embodiment, the image frames of the second (initiating) scene are thus placed on the base layer, at least for the duration of the scene transition. The first image frame 706 of the scene is typically an I-frame, and it is used for temporally predicting the succeeding image frames of the scene. Consequently, the succeeding image frames of the second scene are temporally predicted frames, typically P-frames, such as frames 708 and 710 shown in Figure 7.

[0076] According to a preferred embodiment of the invention, this placing of image frames on scalability layers can be used for implementing a cross-faded scene transition by determining the image layer that is on the base layer always as a background image of maximum opacity (100%), or non-transparency value. During a scene transition, image frames located on enhancement layers are placed onto the background image and their opacity is adjusted for example by means of suitable filters such that the frames gradually change from non-transparent to transparent.

[0077] In the video sequence of Figure 7, there are no image frames on the lower scalability layers during the first base layer image frame 700. For this time instant, the first image frame 700 is only coded into the video sequence.

[0078] The next image frame 706 of the base layer initiates a new (second) scene, during which the image frame 706 is provided with depth positioning that places it as the background image, and its opacity value is set to the maximum. Temporally simultaneously with the image frame 706 of the base layer, there is an image frame 702 of a terminating (first) scene on the first enhancement layer. To allow a cross-faded scene transition to be produced, the transparency of the frame 702 must be increased. The example of Figure 7 assumes that the opacity of the image frame 702 is set at 67% and, in addition, the image frame 702 is provided with depth positioning that determines it as a foreground image. For this time instant, an image combining the image frames 706 and 702 is coded into the video sequence, image 706 being visible as a weaker image on the background and image 702 as a stronger image at the front, because its opacity value is essentially high (67%).

[0079] During the temporally following image frame, there is a second image frame 708 of the second scene on the base layer, the frame

being thus correspondingly provided with depth positioning determining it as a background image, and its opacity value is set to the maximum. The first enhancement layer further comprises the last image frame 704 of a temporally simultaneously terminating (first) scene, the opacity value of the frame being set at 33% and, in addition, the image frame 704 being provided with depth positioning that determines it as a foreground image as well. Consequently, for this time instant, an image combined of the image frames 708 and 704 is coded into the video sequence, the image 708 being displayed as a stronger image on the background and the image 704 as a weaker image on the foreground, because the opacity value of the image 704 is no longer more than 33%.

[0080] During the temporally following image frame, the base layer comprises a third image frame 710 of the second scene. Since the first scene has terminated, only the image frame 710 is coded into the video sequence, and the displaying of the second scene continues from the frame 710.

[0081] The above disclosure describes, by way of example, the positioning of image frames according to the invention on scalability layers to implement cross-faded scene transition in a manner that is advantageous from the point of view of coding efficiency. However, it is possible that when a video sequence is being transmitted or decoded, a situation arises in which the bit rate of the video sequence must be adjusted according to the maximum value of the bandwidth and/or terminal device decoding rate available for data transfer. This kind of bit rate control causes problems when the scene transition is to be implemented using prior art video coding methods.

[0082] A preferred embodiment of the present invention now allows one or more scalability layers, or independently decodable sub-sequences included in them, to be removed from a video sequence, whereby the bit rate of the video sequence can be decreased and yet, at the same time, the video sequence can be decoded without reducing image frequency. In the image frame positioning according to Figure 7, this can be implemented by removing the first enhancement layer from the video sequence. The video sequence is thus only used for displaying the image frames 700, 706, 708 and 710 of the base layer. In other words, a direct transition from the first (terminating) scene to the second (initiating) scene takes place in the form of an abrupt scene transition, i.e. directly from the image frame 700 of the first scene into the I-image frame 706 that initiates the second scene. The transition is thus not a

cross-faded scene transition but an abrupt scene transition. Nevertheless, the scene transition can be carried out in an advantageous manner without affecting the quality of the video sequence image, and the viewer usually does not experience an abrupt scene transition carried out instead of a cross-faded scene transition in any way disturbing or faulty. On the contrary, since the prior art implementation does not allow scalability layers to be removed, scene transition would often require image frequency to be reduced, which the viewer would find jerky and disturbing.

[0083] The invention thus provides a preferred means for carrying out multimedia data traffic shaping in a streaming server comprising information about the different sub-sequences of a video sequence: their average bit rate, location in relation to the entire video sequence, duration and their interdependencies regarding the layers. The streaming server also determines the maximum value of the bandwidth available for the data transfer and/or the decoding rate of the terminal device. On the basis of this information, the streaming server decides how many scalability layers and which sub-sequences are transmitted in the video sequence. Bit rate control can thus be carried out, when necessary, by making first a rough adjustment of the number of the scalability layers, after which finer sub-sequence-specific adjustment can be easily carried out. At its simplest, bit rate control means making sub-sequence-specific decisions on whether a particular sub-sequence will be added to a video sequence or removed from it. In case of removal it is advisable to remove entire sub-sequences from a video sequence, because the removal of separate images may cause errors in other images of the same sub-sequence. For the same reason, all sub-sequences of a lower enhancement layer should be left out if they are dependent on the removed sub-sequence of a higher layer. If there are interdependent sub-sequences on one and the same scalability layer, sub-sequences dependent on an earlier sub-sequence must be removed if the earlier sub-sequence is removed.

[0084] If the image frame identifier data are added to a video sequence that is to be transmitted, traffic shaping can also be carried out in a telecommunications network element to be used for the transfer of the video sequence, for example in an Internet router, in different gateways, or at a base station or base station controller of a mobile communications network. For the network element to be able to maintain and process the sub-sequence information, it must have extra memory and processing capacity. For this

reason traffic shaping that is to be carried out in the network is perhaps most probably executed using simple processing methods, such as the DiffServ, i.e. differentiated services, a procedure that is supported by some IP-based networks. In the DiffServ method, each IP data packet is assigned a priority, whereby data packets of a higher priority are relayed faster and more reliably to the recipient than packets of a lower priority. This is advantageously applied to the scalability of the invention by determining not only scalability-layer-specific, but also sub-sequence-specific priorities, which enables a highly advanced prioritization.

[0085] There are many alternatives for adding image frame identifier data to a video sequence that is to be transmitted. In addition, it is also possible not to include any identifier data into the video sequence, in which case traffic shaping is only carried out at the streaming server. The identifier data can be included in the header fields of a video sequence, or in the header fields of the transfer protocol to be used, such as RTP (Real Time Protocol). In the latter case traffic shaping can be carried out also at one of the network elements of the transfer path. In addition, the receiving terminal device can control the decoding.

[0086] If the encoder or decoder supports reference picture selection, video sequence coding requires that decoded image frames be buffered before the coding so as to allow the relationships between different image frames to be temporally predicted from one or more other image frames. Image frame buffering can be arranged at least in two different ways, either as sliding windowing or as adaptive buffer memory control. In sliding windowing, M image frames that were coded last are used as a buffer. The frames in the buffer memory are in a decoded and reconstructed form, which allows them to be used as reference images in the coding. As the coding proceeds, the image frame buffering functions on the basis of the FIFO principle (First-In-First-Out). Images that are not used as reference images, such as conventional B-images, do not need to be stored in the buffer. Alternatively, the buffering can also be implemented as adaptive buffer memory control, in which case the image buffering is not restricted to the FIFO principle, but image frames that are not needed can be removed from the buffer in the middle of the process, or, correspondingly, some image frames can be stored in the buffer for a longer period of time, if they are needed as reference images for later image frames. A known reference picture selection is implemented by indexing image

frames that are in the buffer memory into a specific order, the image indices being then used to refer to an image in connection with motion-compensation, for example. This indexing method generally provides better compression efficiency compared to using image numbers, for example, for referring to a specific image when motion-compensation reference images are to be signalled.

[0087] The above reference image indexing method is sensitive to transfer errors, because the buffers of the sender's encoder and the recipient's decoder must contain mutually corresponding reconstructed images in identical order to ensure that the encoder and decoder both form the same indexing order. If the image frames are indexed in different order in the buffers of the encoder and the decoder, an incorrect reference image may be used in the decoder. To prevent this, it is essential that the decoder can be controlled to take into account image frames and sub-sequences that the encoder has intentionally removed from the video sequence. In that case the image frame numbering may comprise gaps, which the decoder typically interprets as errors and tries to reconstruct the image frames interpreted as lost. For this reason, it is essential that the encoder is capable to inform the decoder that the discontinuities in the image numbering of the transmitted image frames are intentional.

[0088] In response to this, and provided that sliding windowing is used for buffering the image frames, the decoder enters into the buffer memory a number of image frames, the contents of which may be fully random, corresponding to the missing image numbers. These random image frames are then designated by an identifier "invalid" to indicate that the frames in question do not belong to the actual video sequence, but are only filler frames entered for purposes of buffer memory management. A filler frame can naturally be implemented using only memory indicators, i.e. no data is preferably entered into the buffer memory, but memory management is used merely to store a reference to a generic "invalid" frame. The entering of the image frames of the actual video sequence continues from the correct image frame number after the number of filler frames indicated by the missing image numbers has been entered into the buffer, which allows the buffer memories of the encoder and the decoder to be kept preferably in synchronism. If during decoding a reference to an image number is detected which is then found to indicate a filler frame located in the buffer, error correction actions are initiated

in the decoder to reconstruct the actual reference image, for example by asking the encoder to re-transmit the reference image in question.

[0089] Further, the procedure of the invention allows separate buffer memories to be used on the different scalability layers, or, correspondingly, sub-sequence-specifically. Each scalability layer may thus have a separate buffer memory that is conceptually separate and functions on the basis of the sliding window principle. Similarly, each sub-sequence may also be provided with a conceptually separate buffer memory that also functions on the basis of the sliding window principle. This means that the buffer memory is always emptied when a sub-sequence terminates. Separate buffer memories can be used in a preferred manner for reducing the need for signalling in certain situations in which ordinary sliding window buffering would be inadequate and actively adaptive buffer memory management would need to be used instead.

[0090] The above disclosure describes a procedure for coding video frames for the purpose of producing a scalable, compressed video sequence. The actual procedure is carried out in a video encoder, such as the compressor 108 of Figure 1, which may be any known video encoder. For example a video encoder according to the ITU-T recommendation H.263 or H.26L may be used, the video encoder being arranged to form, in accordance with the invention, a first sub-sequence into a video sequence, at least part of the sub-sequence being formed by coding I-frames; to form at least a second sub-sequence into the video sequence, at least part of the sub -sequence being formed by coding at least P- or B-frames, and at least one video frame of the second sub-sequence being predicted from at least one video frame of the first sub-sequence; and to determine into the video sequence the identification data of at least the video frames of the second sub-sequence.

[0091] According to the procedure of the invention, each sub-sequence of a particular scalability layer is preferably independently decodable, naturally taking into account dependencies from higher scalability layers and possibly other sub-sequences of the same scalability layer. A scalably compressed video sequence such as the one described above can thus be decoded by decoding a first sub-sequence of a video sequence, at least part of the sub-sequence having been formed by coding at least I-frames, and by decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence having been formed by coding at least

P- or B-frames, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence, and by determining the identification and dependency data of at least the video frames comprised by the second sub-sequence of the video sequence, and by reconstructing at least part of the video sequence on the basis of the sub-sequence dependencies.

[0092] The actual decoding takes places in the video decoder, such as the decompressor 118 of Figure 1, which may be any known video decoder. For example, a low bit rate video decoder according to the ITU-T recommendation H.263 or H.26L may be used, which in this invention is arranged to decode a first sub-sequence of a video sequence, at least part of the sub-sequence having been formed by coding I-frames; to decode at least a second sub-sequence of the video sequence, at least part of the second sub-sequence having been formed by coding at least P- or B-frames and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence. The video decoder is arranged to determine the identification and dependency data of at least the video frames comprised by the second sub-sequence of the video sequence and to reconstruct at least part of the video sequence on the basis of the dependencies of the sub-sequences.

[0093] An essential aspect in the operation of the streaming system of the invention is that the encoder and decoder are positioned at least so that the encoder is operationally connected to the streaming server and the decoder is operationally connected to the receiving terminal device. However, the different elements of the streaming system, terminal devices in particular, may include functionalities that allow two-way transfer of multimedia files, i.e. transmission and reception. The encoder and decoder can thus be implemented in the form of what it known as a video codec integrating both encoder and decoder functionalities.

[0094] It is to be noted that according to the invention the functional elements of the above described streaming system and its elements, such as the streaming server, video encoder, video decoder, and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two. The coding and decoding methods of the invention are particularly suitable for implementation as computer software comprising computer-readable commands for executing the process steps of the

invention. A preferred way of implementing the encoder and the decoder is to store them in a storage means as a program code that can be executed by a computer-like device, for example a personal computer (PC) or a mobile station, to provide coding/decoding functionalities on the device in question.

[0095] Another alternative is to implement the invention as a video signal comprising a scalably compressed video sequence which in turn comprises video frames coded according to at least a first and a second frame format, the video frames according to the first frame format being independent of other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames. According to the invention, the video signal in question comprises at least a first sub-sequence, at least part of which has been formed by coding at least video frames of the first frame format; at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format; and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and at least one data field that determines video frames belonging to the second sub-sequence.

[0096] It is apparent to a person skilled in the art that as technology advances the basic idea of the invention can be implemented in various ways. The invention and its embodiments are therefore not restricted to the above examples, but they may vary within the scope of the claims.

CLAIMS

1. A method for coding video frames for the purpose of forming a scalable, compressed video sequence, the compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising:

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

forming into the video sequence information indicating which video frames belong to the second sub-sequence.

2. A method according to claim 1, **characterized** by comprising:

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

3. A method according to claim 2, **characterized** by comprising:

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from a group comprising

a video frame of a higher scalability layer,

an image frame of the second sub-sequence of the same scalability layer.

4. A method according to claim 3, **characterized** by comprising:

determining the dependencies of the video frames of the second sub-sequence on the basis of at least a scalability layer identifier and a sub-sequence identifier.

5. A method according to claim 3 or 4, **characterized** by comprising:

coding the first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

6. A method according to claim 3 or 4, **characterized** by comprising:

coding the first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

7. A method according to claim 3, **characterized** by comprising:

determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.

8. A method according to claim 3, **characterized** by comprising:

determining a unique identifier for each video frame on the basis of the image number.

9. A method according to claim 7 or 8, **characterized** by comprising:

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. A method according to any one of the preceding claims, **characterized** in that

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

11. A method according to any one of the preceding claims, **characterized** by comprising:

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

12. A method according to any one of the preceding claims, **characterized** by comprising:

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

13. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, arranged to transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to control the bit rate of a video sequence transmission in which a first sub-sequence of the video sequence comprises video frames of at least a first frame format and at least a second sub-sequence comprises video frames of at least a second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

the bit rate control being arranged to be carried out in at least one of the following ways:

at least one sub-sequence is removed from the video sequence,

at least one sub-sequence is added to the video sequence.

14. A video encoder for forming a scalable, compressed video sequence, the compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format; and to

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and to

form into the video sequence information indicating which video frames belong the second sub-sequence.

15. A method for decoding a scalably compressed video sequence, the compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame

format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising:

decoding a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;

decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determining dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and

reconstructing at least part of the video sequence on the basis of the dependencies of the sub-sequences.

16. A method according to claim 15, wherein video frames are entered into a gliding buffer memory in connection with decoding, **characterized** by comprising:

decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;

configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

17. A method according to claim 16, **characterized** by comprising:

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

18. A method according to any one of claims 15 to 17, **characterized** by comprising:

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

19. A method according to any one of claims 15 to 18, **characterized** by comprising:

initiating the decoding from a random point in the video sequence;

determining an independently decodable sub-sequence that is next after the random point, and

setting the value of the image number of the first video frame in the sub-sequence at zero.

20. A video decoder for decoding a scalably compressed video sequence, the compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video decoder is arranged to

decode a first sub-sequence of the video sequence, at least part of which has been formed by coding at least video frames of the first frame format;

decode at least a second sub-sequence of the video sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.

21. A computer program for coding video frames so as to form a scalable, compressed video sequence, the compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining data into the video sequence to determine which video frames belong to the second sub-sequence.

22. A computer program for decoding a scalably compressed video sequence, the video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the computer program comprises

a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.

23. A video signal comprising a scalably compressed video sequence, the video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video signal comprises

a first sub-sequence, at least part of which has been formed by coding at least video frames of the first frame format;

at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at

least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

at least one data field determining the video frames that belong to the second sub-sequence.

(57) ABSTRACT

A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format. The video frames of the first frame format are independent of the other video frames, and the video frames of the second frame format are predicted from at least one of the other video frames. The video sequence is provided with a first sub-sequence determined therein, at least part of the first sub-sequence having been formed by coding at least video frames of the first frame format; and with at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence. Identifier data of at least the video frames comprised by the second sub-sequence is determined into the video sequence.

(Figure 2)

P.B.5818 - Patentlaan 2
2280 HV Rijswijk (ZH)
☎ +31 70 340 2040
TX 31651 epo nl
FAX +31 70 340 3016

Europäisches Patentamt | **European Patent Office** | **Office européen des brevets**

Eingangsstelle | Receiving Section | Section de Dépôt

Kaukonen, Juha Veikko
Kolster Oy Ab,
Iso Roobertinkatu 23,
P.O. Box 148
00121 Helsinki
FINLANDE

Datum/Date
13/10/04

Zeichen/Ref./Réf.	Anmeldung Nr./Application No./Demande n°./Patent Nr ./Patent No./Brevet n°.
2011658	03700321.7-2223 / 1479245

Anmelder/Applicant/Demandeur/Patentinhaber/Proprietor/Titulaire
Nokia Corporation

NOTIFICATION OF EUROPEAN PUBLICATION NUMBER AND INFORMATION ON THE APPLICATION OF ARTICLE 67(3) EPC

The provisional protection under Article 67(1) and (2) EPC in the individual Contracting States becomes effective only when the conditions referred to in Article 67(3) EPC have been fulfilled (for further details, see information brochure of the European Patent Office "National Law relating to the EPC" and additional information in the Official Journal of the European Patent Office).

Pursuant to Article 158(1) EPC the publication under Article 21 PCT of an international application for which the European Patent Office is a designated Office takes the place of the publication of a European patent application.

The bibliographic data of the above-mentioned Euro-PCT application will be published on 24.11.04 in Section I.1 of the European Patent Bulletin.
The European publication number is 1479245.

In all future communications to the European Patent Office, please quote the application number plus Directorate number.

RECEIVING SECTION

P.B.5818 - Patentlaan 2
2280 HV Rijswijk (ZH)
(070) 3 40 20 40
FAX (070) 3 40 30 16

Europäisches Patentamt | **European Patent Office** | **Office européen des brevets**

Generaldirektion 1 | Directorate General 1 | Direction générale 1

Kaukonen, Juha Veikko
Kolster Oy Ab,
Iso Roobertinkatu 23,
P.O. Box 148
00121 Helsinki
FINLANDE

epoline® Customer Services
Tel.: +31 (0)70 340 45 00

Date 01-10-2004

Reference	Application No./Patent No.
2011658	03700321.7 - 2223 PCT/FI0300053

Applicant/Proprietor
Nokia Corporation

Communication pursuant to Rules 109 and 110 EPC

(1) Amendment of application documents, especially the claims (Rule 109 EPC)

The above mentioned international (Euro-PCT) application has entered the European phase, or can do so, once the necessary conditions are fulfilled.

Under Articles 28, 41 PCT, Rules 52, 78 PCT and Rule 86(2) to (4) EPC, the applicant may amend the application documents after receiving the international search report.

Whether or not he has already done so, he now has a further opportunity to file amended claims or other application documents within a non-extendable time limit of ONE MONTH after notification of the present communication (Rule 109 EPC).

The claims applicable on expiry of the above time limit, i.e. those filed on entry into the European phase or in response to the present communication, will form the basis for the calculation of any claims fee to be paid (see page 2) and for any supplementary search to be carried out under Article 157(2) EPC (Rule 109 EPC).

--2/

(2) Claims fees under Rule 110 EPC

If the application documents on which the European grant procedure is to be based comprise more than ten claims, a claims fee shall be payable for the eleventh and each subsequent claim within the period provided for in Rule 107(1) EPC.

☑ Based on the application documents currently on file, all necessary claims fees have already been paid (or the documents do not comprise more than 10 claims).

☐ All necessary fees will be/have been debited automatically according to the automatic debit order.

☐ The claims fee due for the claims ..... to ..... were not paid within the above-mentioned period.

Any non-paid claims fee, either based on the current set of claims or on any amended claims to be filed pursuant to Rule 109 EPC (see page 1), may still be validly paid within a non-extendable period of grace of **ONE MONTH** after notification of this communication.

If a payment is made for only some of the claims, it must be indicated for which claims it is intended. If a claims fee is not paid in due time, the claim concerned is deemed to be abandoned (Rule 110(4) EPC).

If claims fees have already been paid, but on expiry of the above-mentioned time limit there is a new set of claims containing fewer fee-incurring claims than previously, the claims fees in excess of those due under Rule 110(2), 2nd sentence, EPC will be refunded (Rule 110(3) EPC).

You are reminded that any supplementary search under Article 157(2) EPC will relate only to the last set of claims applicable on expiry of the above time limit AND will be confined to those fee-incurring claims for which fees have been paid in due time.

The fee for the eleventh and each subsequent claim is EUR 40,00.

RECEIVING SECTION
Van Laar-Rabelink, Bertie

Europäisches Patentamt • European Patent Office • Office européen des brevets

Acknowledgement of receipt

We hereby acknowledge receipt of the form for entry into the European phase (EPO as designated or elected Office) as follows:

Submission number	22883	
PCT application number	PCT/FI03/00053	
Date of receipt	29 July 2004	
Your reference	2011658	
Applicant		
Country		
Documents submitted	EPF1200.PDF OTHER-1.PDF ep-euro-pct.xml	OTHER-0.PDF application-body.xml package-data.xml
Submitted by	CN=Antero Virkkala,O=Kolster Oy AB,C=FI	
Method of submission	Online	
Date and time receipt generated	29 July 2004, 10:20:45	
Digest	84:96:42:F0:5E:F8:9D:7A:68:6D:BE:91:ED:39:31:C2:81:1B:28:E1	

/European Patent Office/

PATENT COOPERATION TREATY

From the
INTERNATIONAL PRELIMINARY EXAMINING AUTHORITY

To:

Kolster OY AB
Iso Roobertinkatu 23,
P.O. Box 148
FIN-00121 HELSINKI

PCT

WRITTEN OPINION

(PCT Rule 66)

FRIST: 9.2.04 inf. 15/12-03 pir

Date of mailing *(day/month/year)*	11-12-2003
Applicant's or agent's file reference: 2011658PC/ko	**REPLY DUE** within 60 days from the above date of mailing

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/FI03/00053	22-01-2003	23-01-2002

International Patent Classification (IPC) or both national classification and IPC7
H04N 7/50

Applicant
Nokia Corporation et al.

1. This written opinion is the first (first, etc.) drawn by this International Preliminary Examining Authority.
2. This opinion contains indications relating to the following items:
   - I [X] Basis of the report
   - II [ ] Priority
   - III [ ] Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
   - IV [ ] Lack of unity of invention
   - V [X] Reasoned statement under Rule 66.2(a)(ii) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
   - VI [ ] Certain documents cited
   - VII [ ] Certain defects in the international application
   - VIII [ ] Certain observations on the international application
3. The applicant is hereby **invited to reply** to this opinion.

   When? See the time limit indicated above. The applicant may, before the expiration of that time limit, request this Authority to grant an extension, see Rule 66.2(d).

   How? By submitting a written reply, accompanied, where appropriate, by amendments, according to Rule 66.3. For the form and the language of the amendments, see Rules 66.8 and 66.9.

   Also For an additional opportunity to submit amendments, see Rule 66.4. For the examiner's obligation to consider amendments and/or arguments, see Rule 66.4*bis*. For an informal communication with the examiner, see Rule 66.6.

   If no reply is filed, the international preliminary examination report will be established on the basis of this opinion.
4. The final date by which the international preliminary examination report must be established according to Rule 69.2 is: 23-05-2004.

Name and mailing address of the IPEA/SE	Authorized officer
Patent- och registreringsverket Box 5055 S-102 42 STOCKHOLM Telex 17978 PATOREG-S Facsimile No. 08-667 72 88	Alexander Lakic /LR Telephone No. 08-782 25 00

Form PCT/IPEA/408 (cover sheet) (January 1998)

→ Pg 15/12-03 pir

WRITTEN OPINION

International application No.
PCT/FI03/00053

I. Basis of the opinion

1. With regard to the **elements** of the international application:*

☒ the international application as originally filed

☐ the description:
pages _____, as originally filed
pages _____, filed with the demand
pages _____, filed with the letter of _____

☐ the claims:
pages _____, as originally filed
pages _____, as amended (together with any statement) under article 19
pages _____, filed with the demand
pages _____, filed with the letter of _____

☐ the drawings:
pages _____, as originally filed
pages _____, filed with the demand
pages _____, filed with the letter of _____

☐ the sequence listing part of the description:
pages _____, as originally filed
pages _____, filed with the demand
pages _____, filed with the letter of _____

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.
These elements were available or furnished to this Authority in the following language _____ which is:

☐ the language of a translation furnished for the purposes of international search (under Rule 23.1(b)).

☐ the language of publication of the international application (under Rule 48.3(b)).

☐ the language of the translation furnished for the purposes of international preliminary examination (under Rules 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the written opinion was drawn on the basis of the sequence listing:

☐ contained in the international application in printed form.

☐ filed together with the international application in computer readable form.

☐ furnished subsequently to this Authority in written form.

☐ furnished subsequently to this Authority in computer readable form.

☐ The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

☐ The statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

4. ☐ The amendments have resulted in the cancellation of:

☐ the description, pages _____

☐ the claims, Nos. _____

☐ the drawings, sheet/fig _____

5. ☐ This opinion has been drawn as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed, as indicated in the Supplemental Box (Rule 70.2 (c)).

* *Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this opinion as "originally filed".*

V. Reasoned statement under Rule 66.2(a)(ii) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. Statement

Novelty (N)	Claims	1-24	YES
	Claims		NO
Inventive step (IS)	Claims		YES
	Claims	1-24	NO
Industrial applicability (IA)	Claims	1-24	YES
	Claims		NO

2. Citations and explanations

Documents cited in the International Search Report:

D1: WO 0147283 A1
D2: US 6337881 B1
D3: EP 1005232 A2
D4: US 5774593 A
D5: US 6307886 B1

D1 is considered to the most relevant document.

D1 discloses a system for determining whether simulcast coding or temporal/spatial scalability coding should be used to encode video for clients with a specific communication link. The part which concerns the spatial and temporal scalability coding is considered to be the most relevant part of D1, because the invention according to the application concerns this kind of coding technique.

The invention according to the application concerns a method for coding video frames. The aim of the invention is to code the video frames into a scalable a compressed video sequence. The scalability feature can be utilized when streaming a video sequence to multiple clients, where clients have different amount of bandwidth or processing power. The method, according to the claimed invention, includes a way of dividing a video sequence into sub-sequences, where each sub-sequence can be decoded independently. In the video sequence there is also an indication included which shows which frames belong to a certain sub-sequence. According to the application, the above mentioned indication facilitates inclusion of extra material, e.g. a commercial clip, into the video sequence.

.../...

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

Continuation of: Box V

Referring to claim 1, 13-15, 21-24:

Claim 1 reveals a method for coding video frames into a scalable and compressed video sequence. The claim also includes that the video sequence comprise two frame formats (which are later in the claims identified as I- and P-frames). Claim 1 further describes that into the video sequence two sub-sequences are formed, where at least one video frame of the second sub-sequence has been predicted from at least on video frame from the first sub-sequence. Information indicating which video frame belongs to the second sub-sequence is formed into the video sequence.

In D1 (see figure 2 and page 13, row 21, to page 14, row 5) temporal scalability is illustrated and described. As can be seen from figure 2, the layers comprise I-, P- and B-frames. From the figure it can also be seen that the video sequence is composed of several sub-sequences.

The invention according to claim 1 differs by mentioning a way of including, into the video stream, an indication of which video frame belongs to the second sub-sequence. Although the above difference is not explicitly mentioned in D1, it is considered to be an obvious measure for a person skilled, facing the previous described problem, to include, in the video stream, some kind of information that indicates which frames belong to a certain sub-sequence.
Therefore, the invention according to claim 1 is not considered to involve an inventive step.

The independent claims 13-15, 21-24 are based on the same idea that is presented in D1. Although the above claims represent different categories (e.g. a decoder, a computer program), the argumentation regarding claim 1 is considered to be valid for claims 13-15 and 21-24 as well. Therefore, the invention of claims 13-15 and 21-24 is not considered to involve an inventive step.

Referring to claims 2-12, 16-20:

All the additional features described in the above mentioned claims are either known from the prior art documents cited in the International Search Report or generally known in the technical field of video processing/compression. Therefore, the invention according to claims 2-12, 16-20 is not considered to involve an inventive step.

KOLSTER

PATENT- OCH
REGISTRERINGSVERKET
VALHALLAVÄGEN 36
BOX 5055
S-102 42 STOCKHOLM 5
SVERIGE

-9 -02- 2004

Attn: Mr. Alexander Lakic / LR

Fax: +46 8 667 72 88 (2 pages), original by mail

February 6, 2004

KOLSTER OY AB
Pääkonttori
Iso Roobertinkatu 23
PL 148
00121 Helsinki
Huvudkontor
Stora Robertsgatan 23
PB 148
00121 Helsingfors
Head office
Iso Roobertinkatu 23
PO Box 148
FIN-00121 Helsinki
Finland
Tel. +358 (0)9 618 821
Fax +358 (0)9 602 244
kolster@kolster.fi

Perustettu 1874
Y-tunnus 0110129-2
Kotipaikka Helsinki
Grundad 1874
FO-nummer 0110129-2
Hemort Helsingfors
Established 1874
Business ID 0110129-2
Reg. Office Helsinki

Helsinki/Helsingfors
Turku/Åbo
Tampere/Tammerfors
Vaasa/Vasa
Oulu/Uleåborg

www.kolster.com

INTERNATIONAL PATENT APPLICATION No. PCT/FI03/00053
Applicant: Nokia Corporation
Our Ref: 2011658PC/Pg/art

Dear Mr. Lakic,

In response to the Written Opinion, dated 11.12.2003, regarding the PCT application mentioned above, the Applicant considers the originally filed claims to involve both novelty and an inventive step. The Applicant submits the following arguments:

Novelty

The Examiner has not raised any objections regarding the novelty of the originally filed claims.

Inventive Step

The Examiner has cited one document, WO01/47283, which, according to the Examiner, teaches a skilled man such that the present would become obvious for him.

WO01/47283 discloses a method for coding a scalable multilayer video sequence, wherein the layers include a base layer and one or more enhancements layers. By adjusting the number of the layers, the frame rate, resolution and the quality of the video sequence can be affected, whereby the available bandwidth can be the decisive criterium. Furthermore, WO01/47283 discloses how I-, P- and B-frames are positioned on different layers forming several group of pictures (GOPs). There is no dependencies between the group of pictures, but only inside each group of pictures.

Thus, WO01/47283 discloses a method, which is stated as prior art for the present application (chap. 0009 – 0010). Therefore, also the drawbacks of the prior art mentioned in the present application are included in the solution according to WO01/47283: carrying out reference picture selection when the prediction includes images temporally succeeding the image to be coded, how to signal differences of significance between INTER-frames, and how to create a frame numbering of video

sequence, which allows flexible the insertion of a video sequence in the middle of another video sequence and scaling of a video sequence.

The problem is solved by a method for coding video frames for the purpose of forming a scalable, compressed video sequence, wherein a video signal is formed, which video signal comprises a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format; at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and at least one data field determining the video frames that belong to the second sub-sequence.

Hence, the idea underlying the present invention is to create a scalable coding hierarchy, wherein, for each sub-sequence, all sub-sequences are determined, upon which the coding of said sub-sequence is dependent. This is achieved by a frame numbering hierarchy, wherein the frame numbering of each sub-sequence includes information on all sub-sequences, which have been used in image frame prediction. This information is signalled along the video sequence, but separately of the image information. This enables a flexible scaling of video sequence, since independently decodable parts of the video sequence can easily be determined.

WO01/47283 would not give a man skilled in the art any hints for such a solution. First of all, in WO01/47283 there can be no dependencies between the group of pictures; therefore the group of pictures is not a sub-sequence as it is determined in the present application. Secondly, WO01/47283 does not include any information how the frame numbering of the video sequence should be carried out. It is commonly known in the prior art to use ascending arithmetical frame numbering, which most probably is used in the solution of WO01/47283 as well, although not expressly stated. Such a frame numbering, however, cannot be used to solve problems underlying the present invention.

Therefore is respectfully noted that the invention according to the independent claims 1, 13 - 15 and 21 - 24 is considered both novel and inventive.

Yours faithfully

KOLSTER OY AB

Eero Palmgren

To the European Patent Office

Entry into the European phase (EPO as designated or elected Office)

European application number	EP03700321.7
PCT application number	PCT/FI2003/00053
PCT publication number	WO03063505
Applicant's or representative's reference	2011658

1. Applicant

Particulars of the applicant(s) are contained in the international publication or were recorded by the International Bureau subsequent to the international publication. ☑

Changes which have not yet been recorded by the International Bureau are set out here: ☐

Address for correspondence

2. Representative 1

This is the representative who will be listed in the Register of European Patents and to whom notifications will be made

Name Kaukonen, Mr Juha

Address of place of business

Kolster Oy Ab
(Iso Roobertinkatu 23) P.O.Box 148
FIN-00121 Helsinki
Finland

Telephone +358-9-618 821

Fax +358-9-602 244

e-mail epline@kolster.fi

Any additional representative(s) is/are listed here: ☐

3. Authorisation

An individual authorisation is attached. ☐

A general authorisation has been registered under No: ☐

A general authorisation has been filed, but not yet registered. ☐

The authorisation filed with the EPO as PCT receiving Office expressly includes the European phase. ☐

4. Request for examination

Examination of the application under Art. 94 EPC is hereby requested. The examination fee is being (has been, will be) paid. ☑

Request for examination in an admissible non-EPO language: ☑

Täten pyydetään hakemuksen tutkimista artiklan 94 mukaisesti.

5. Copies

One or more additional sets of copies of the documents cited in the supplementary European search report are hereby requested. ☐

Number of additional sets of copies

6. Documents intended for proceedings before the EPO

6.1 Proceedings before the EPO as designated Office (PCT I) are to be based on the following documents:

the application documents published by the International Bureau (with all claims, description and drawings), where applicable with amended claims under Art. 19 PCT ☑

unless replaced by the amendments attached. ☐

Where necessary, clarifications should be attached as 'Other Documents'

6.2 Proceedings before the EPO as elected Office (PCT II) are to be based on the following documents:

the documents on which the international preliminary examination report is based, including any annexes ☑

unless replaced by the amendments attached. ☐

Where necessary, clarifications should be attached as 'Other Documents'

If the EPO as International Preliminary Examining Authority has been supplied with test reports, these may be used as the basis of proceedings before the EPO. ☑

7. Translations

Translations in one of the official languages of the EPO (English, French, German) are attached as crossed below:

* *In proceedings before the EPO as designated or elected Office (PCT I + II):*

Translation of the international application (description, claims, any text in the drawings) as originally filed, of the abstract as published and of any indication under Rule 13bis.3 and 13bis.4 PCT regarding biological material ☐

Translation of priority application(s) ☐

It is hereby declared that the international application as originally filed is a complete translation of the previous application (Rule 38(5) EPC) ☐

* *In addition, in proceedings before the EPO as designated Office (PCT I):*

Translation of amended claims and any statement under Art. 19 PCT, if the claims as amended are to form the basis for the proceedings before the EPO (see Section 6). ☐

* *In addition, in proceedings before the EPO as elected office (PCT II):*

Translation of annexes to the international preliminary examination report ☐

8. Biological material

The invention relates to and/or uses biological material deposited under Rule 28 EPC. ☐

The particulars referred to in Rule 28(1)(c) EPC (if not yet known, the depository institution and the identification reference(s) [number, symbols, etc.] of the depositor) are given in the international publication or in the translation submitted under Section 7 on: ☐

page(s) / line(s)

A copy of the receipt(s) of deposit issued by the depositary institution

is attached ☐

will be filed at a later date ☐

A waiver of the right to an undertaking from the requester pursuant to Rule 28(3) EPC is attached. ☐

9. Nucleotide and amino acid sequences

The items required under Rules 5.2 and 13ter PCT and Rule 111(3) EPC have already been furnished to the EPO. ☐

The sequence listing as part of the description is attached in PDF format. ☐

The sequence listing does not include matter that goes beyond the content of the application as filed. ☐

In addition, the sequence listing data is attached in computer-readable form in accordance with WIPO Standard 25. ☐

The sequence listing data in computer-readable form in accordance with WIPO Standard 25 is identical to the sequence listing in PDF format. ☐

10. Designation fees

10.1 It is currently intended to pay seven times the amount of the designation fee. The designation fees for all the EPC contracting states designated in the international application are thereby deemed to have been paid (Art. 2 No. 3 RFees). ☑

AT BE BG CH&LI CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL PT SE SI SK TR

10.2 The declaration in No. 10.1 does not apply. Instead, it is currently intended to pay fewer than seven designation fees for the following EPC contracting states designated in the international application: ☐

It is requested that no communications under Rule 108(3) EPC be issued in respect of any contracting states not indicated.

10.3 If an automatic debit order has been issued (Section 12), the EPO is authorised, on expiry of the basic period under Rule 107(1)(d) EPC, to debit seven times the amount of the designation fee. If states are indicated in No. 10.2, the EPO will debit designation fees for those states only, unless instructed otherwise before the basic period expires. ☑

11. Extension of the European patent

This application is also considered as being a request for extension to all the non-contracting states to the EPC designated in the international application with which "extension agreements" were in force on the date of filing the international application. However, the extension only takes effect if the prescribed extension fee is paid. ☑

It is currently intended to pay the extension fee for the following states:

12. Debit from deposit account ☑

Currency: EUR

The European Patent Office is hereby authorised to debit from the following deposit account any fees and costs indicated on the fees page.

Deposit account number: 28150002
Account holder: Kolster Oy Ab

13. Reimbursements (if any) should be made to the following EPO deposit account: ☑

Number and account holder: 28150002, Kolster Oy Ab

14. Fees

		Factor applied	Fee schedule	Amount to be paid
14-1	002 Search fee	0	690.00	0.00
14-2	005 Designation fee	7	75.00	525.00
14-3	006 Examination fee	1 (80%)	1 430.00	1 144.00
14-4	015 Claims fee	14	40.00	560.00
14-5	020 Basic national fee for an international application	1	90.00	90.00
	Total:		**EUR**	**2 319.00**

15. Annotations

15-1. Note (for EPO) (EP Phase)

Change of inventor's address (Virkkala, Mr. Antero; 26.7.2004)
Current address for inventor: Miska Hannuksela, Lakotie 15, FI-36110 Ruutana

16. Signature(s) of applicant(s) or representative

Place: **Helsinki**
Date: **29 July 2004**
Signed by: **Subject: FI, Kolster Oy AB, Antero Virkkala**
Issuer: DE, D-Trust GmbH, D-Trust for EPO 2.0
Capacity: **(Representative)**

For employees (Art. 133(3) EPC) having a general authorisation:
General authorisation No.

P.B.5818 - Patentlaan 2
2280 HV Rijswijk (ZH)
(070) 3 40 20 40
FAX (070) 3 40 30 16

Europäisches Patentamt	European Patent Office	Office européen des brevets
Generaldirektion 1	Directorate General 1	Direction générale 1

KOLSTER OY AB
Iso Roobertinkatu 23
P.O. Box 148
FIN-00121 Helsinki
FINLANDE

epoline® Customer Services
Tel.: +31 (0)70 340 45 00

Date
07.06.04

Reference	Application No./Patent No.
	03700321.7 - 2223 PCT/FI0300053
Applicant/Proprietor	
Nokia Corporation	

Entry into the European phase before the European Patent Office

These notes describe the procedural steps required for entry into the European phase before the European Patent Office (EPO). You are advised to read them carefully: failure to take the necessary action in time can lead to your application being deemed withdrawn.

1. The above-mentioned international patent application has been given European application No. **03700321.7**.

2. Applicants **without** a residence or their principal place of business in an EPC contracting state may themselves initiate European processing of their international applications, provided they do so before expiry of the 31st month from the priority date (see also point 6 below).

   During the European phase before the EPO as designated or elected Office, however, such applicants must be represented by a professional representative (Arts. 133(2) and 134(1), (7) EPC).

   Procedural acts performed after expiry of the 31st month by a professional representative who acted during the international phase but is not authorised to act before the EPO have no legal effect and therefore lead to loss of rights.

   Please note that a professional representative authorised to act before the EPO and who acted for the applicant during the international phase does not automatically become the representative for the European phase. Applicants are therefore strongly advised to appoint in good time any representative they wish to initiate the European phase for them; otherwise, the EPO has to send all communications direct to the applicant.

3. Applicants **with** a residence or their principal place of business in an EPC contracting state are not obliged to appoint, for the European phase before the EPO as designated or elected Office, a professional representative authorised to act before the EPO.
   However, in view of the complexity of the procedure it is recommended that they do so.

4. Applicants and professional representatives are also strongly advised to initiate the European phase using EPO Form 1200 (available free of charge from the EPO). This however is not compulsory.

5. **To enter the European phase before the EPO**, the following acts must be performed.
(N.B.: Failure validly to do so will entail loss of rights or other adverse legal consequences.)

5.1 If the EPO is acting as **designated** or **elected** Office (Arts. 22(1)(3) and 39(1) PCT respectively), applicants must, within 31 months from the date of filing or (where applicable) the earliest priority date:

a) Supply a translation of the international application into an EPO official language, if the International Bureau did not publish the application in such a language (Art. 22(1) PCT and Rule 107(1)(a) EPC).
If the translation is not filed in time, the international application is deemed withdrawn before the EPO (Rule 108(1) EPC).
This loss of rights is deemed not to have occurred if the translation is then filed within a two-month grace period as from notification of an EPO communication, provided a surcharge is paid at the same time (Rule 108(3) EPC).

b) Pay the national basic fee (EUR 125,00) and, where a supplementary European search report has to be drawn up, the search fee (EUR 690,00 ; Rule 107(1)(c) and (e) EPC).

c) If the time limit under Article 79(2) EPC expires before the 31-month time limit, pay the designation fee (EUR 75,00) for each contracting state designated (Rule 107(1)(d) EPC).

d) If the time limit under Article 94(2) EPC expires before the 31-month time limit, file the written request for examination **and** pay the examination fee (EUR 1430,00 ; Rule 107(1)(f) EPC).

e) Pay the third-year renewal fee (EUR 380,00) if it falls due before expiry of the 31-month time limit (Rule 107(1)(g) EPC).

If the fees under (b) to (d) above are not paid in time, or the written request for examination is not filed in time, the international application is deemed withdrawn before the EPO, or the contracting-state designation(s) in question is (are) deemed withdrawn (Rule 108(1) and (2) EPC). However, the fees may still be validly paid within a two-month grace period as from notification of an EPO communication, provided the necessary surcharges are paid at the same time (Rule 108(3) EPC). For the renewal fee under (e) above, the grace period is **six** months from the fee's due date (Article 86(2) EPC).

5.2 If the application documents on which the European grant procedure is to be based comprise more then ten claims, a claims fee is payable within the 31-month time limit under Rule 107(1) EPC for the eleventh and each subsequent claim (Rule 110(1) EPC). The fee can however still be paid within a one-month grace period as from notification of an EPO communication pointing out the failure to pay (Rule 110(2) EPC).

6. If the applicant had a representative during the application's international phase, the present notes will be sent to the representative, asking him to inform the applicant accordingly.

All subsequent communications will be sent to the applicant, or - if the EPO is informed of his appointment in time - to the applicant's European representative.

7. For more details about time limits and procedural acts before the EPO as designated and elected Office, see the EPO brochure

   How to get a European patent
   Guide for applicants - Part 2
   PCT procedure before the EPO - "Euro-PCT"

   This brochure, the list of professional representatives before the EPO, Form 1200 and details of the latest fees are now all available on the Internet under

   http://www.european-patent-office.org

RECEIVING SECTION

EPO - DG

24. 05. 2004

(104)

PATENT COOPERATION TREATY

PCT

REC'D 27 APR 2004

WIPO PCT

INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY

(Chapter II of the Patent Cooperation Treaty)

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference	FOR FURTHER ACTION See Form PCT/IPEA/416	
2011658PC/ko		
International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/FI 2003/000053	22-01-2003	23-01-2002
International Patent Classification (IPC) or national classification and IPC		
H04N 7/50		
Applicant		
Nokia Corporation et al		

1. This report is the international preliminary examination report, established by this International Preliminary Examining Authority under Article 35 and transmitted to the applicant according to Article 36.
2. This REPORT consists of a total of 4 sheets, including this cover sheet.
3. This report is also accompanied by ANNEXES, comprising:

   a. ☐ *(sent to the applicant and to the International Bureau)* a total of _____ sheets, as follows:

   ☐ sheets of the description, claims and/or drawings which have been amended and are the basis of this report and/or sheets containing rectifications authorized by this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions).

   ☐ sheets which supersede earlier sheets, but which this Authority considers contain an amendment that goes beyond the disclosure in the international application as filed, as indicated in item 4 of Box No. I and the Supplemental Box.

   b. ☐ *(sent to the International Bureau only)* a total of (indicate type and number of electronic carrier(s)) _____, containing a sequence listing and/or tables related thereto, in computer readable form only, as indicated in the Supplemental Box Relating to Sequence Listing (see Section 802 of the Administrative Instructions).

4. This report contains indications relating to the following items:

☒	Box No. I	Basis of the report
☐	Box No. II	Priority
☐	Box No. III	Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
☐	Box No. IV	Lack of unity of invention
☒	Box No. V	Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
☐	Box No. VI	Certain documents cited
☐	Box No. VII	Certain defects in the international application
☐	Box No. VIII	Certain observations on the international application

Date of submission of the demand	Date of completion of this report
04-08-2003	19-04-2004
Name and mailing address of the IPEA/SE	Authorized officer
Patent- och registreringsverket Box 5055 S-102 42 STOCKHOLM Facsimile No. +46 8 667 72 88	Alexander Lakic /LR Telephone No. +46 8 782 25 00

Form PCT/IPEA/409 (cover sheet) (January 2004)

Copied from 03/02/20 on 08-07-2004

INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY

International application No.
PCT/FI 2003/000053

Box No. I Basis of the report

1. With regard to the **language**, this report is based on the international application in the language in which it was filed, unless otherwise indicated under this item.

☐ This report is based on a translation from the original language into the following language _____ , which is the language of a translation furnished for the purposes of:

- ☐ international search (under Rules 12.3 and 23.1(b))
- ☐ publication of the international application (under Rule 12.4)
- ☐ international preliminary examination (under Rules 55.2 and/or 55.3)

2. With regard to the **elements** of the international application, this report is based on *(replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to this report):*

☒ the international application as originally filed/furnished

☐ the description:

pages _____ as originally filed/furnished
pages* _____ received by this Authority on _____
pages* _____ received by this Authority on _____

☐ the claims:

pages _____ as originally filed/furnished
pages* _____ as amended (together with any statement) under Article 19
pages* _____ received by this Authority on _____
pages* _____ received by this Authority on _____

☐ the drawings:

pages _____ as originally filed/furnished
pages* _____ received by this Authority on _____
pages* _____ received by this Authority on _____

☐ a sequence listing and/or any related table(s) – see Supplemental Box Relating to Sequence Listing.

3. ☐ The amendments have resulted in the cancellation of:

- ☐ the description, pages _____
- ☐ the claims, Nos. _____
- ☐ the drawings, sheets/figs _____
- ☐ the sequence listing *(specify)*: _____
- ☐ any table(s) related to the sequence listing *(specify)*: _____

4. ☐ This report has been established as if (some of) the amendments annexed to this report and listed below had not been made, since they have been considered to go beyond the disclosure as filed, as indicated in the Supplemental Box (Rule 70.2(c)).

- ☐ the description, pages _____
- ☐ the claims, Nos. _____
- ☐ the drawings, sheets/figs _____
- ☐ the sequence listing *(specify)*: _____
- ☐ any table(s) related to the sequence listing *(specify)*: _____

* *If item 4 applies, some or all of those sheets may be marked "superseded."*

Copied from 03/02/2004 on 08-07-2004

INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY

International application No.
PCT/FI 2003/000053

Box No. V **Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement**

1. Statement

Novelty (N)	Claims	1-24	YES
	Claims		NO
Inventive step (IS)	Claims	1-24	YES
	Claims		NO
Industrial applicability (IA)	Claims	1-24	YES
	Claims		NO

2. Citations and explanations (Rule 70.7)

Documents cited in the International Search Report:

D1: WO 0147283 A1
D2: US 6337881 B1
D3: EP 1005232 A2
D4: US 5774593 A
D5: US 6307886 B1

D1 is considered to the most relevant document. D2-D5 only disclose the general state of the art, and are not considered being of particular relevance.

D1 discloses a system for determining whether simulcast coding or temporal/spatial scalability coding should be used to encode video for clients with a specific communication link. The part which concerns the spatial and temporal scalability coding is considered to be the most relevant part of D1.

The invention according to the application concerns a method for coding video frames. The aim of the invention is to code the video frames into a scalable and compressed video sequence. The scalability feature can be utilized when streaming a video sequence to multiple clients, where clients have different amount of bandwidth or processing power. The method, according to the claimed invention, includes a way of dividing a video sequence into sub-sequences, where each sub-sequence can be decoded independently. In the video sequence there is also an indication included which shows which frames belong to a certain sub-sequence. According to the application, the above mentioned indication facilitates inclusion of extra material, e.g. a commercial clip, into the video sequence.

.../...

Copied from 03700320 on 08-07-2004

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.
Continuation of: Box V

Referring to claim 1-24:

Claim 1-24 reveals a method for coding video frames into a scalable and compressed video sequence. The claims also includes that the video sequence comprise two frame formats (I- and P-frames). Claim 1-24 further describes that into the video sequence, two sub-sequences are formed, where at least one video frame of the second sub-sequence has been predicted from at least on video frame from the first sub-sequence. Information indicating which video frame belongs to the second sub-sequence is formed into the video sequence.

In D1 (see figure 2 and page 13, row 21, to page 14, row 5) temporal scalability is illustrated and described. As can be seen from figure 2, the layers comprise I-, P- and B-frames. From the figure it can also be seen that the video sequence is composed of several GOP:s (Group of Pictures).

The invention according to claims 1-24 differs from D1 in the way the scalable coding hierarchy is structured. The multilayer sequence presented in D1 only defines dependencies within each GOP and no dependencies are defined between the GOP:s, therefore the GOP:s mentioned in D1 do not correspond to the sub-sequences according to the applied invention. The sub-sequences defined in claims 1-24 however, are independently decodable because the frame numbering of each sub-sequence includes information on all sub-sequences that has been used in the image frame prediction. A person skilled in the art with aid of the information in D1 would not be able to arrive at the solution according to claims 1-24, because there are no hints of how to modify the coding hierarchy defined in D1 in order to achieve dependencies between the GOP:s. Also, no suggestions or hints are given in D1 on how the frame numbering of the video sequence is carried out.

In view of the aforementioned, the invention according to claims 1-24 is novel, is considered to involve an inventive step and has industrial applicability.

Copied from 32026020 on 08-07-2004

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI 03/00053

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04N 7/50
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04N, H04L, H04M, G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA, PAJ, INSPEC, TDB, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 0147283 A1 (GENERAL INSTRUMENTS CORP), 28 June 2001 (28.06.01), page 11, line 10 - line 20; page 12, line 5 - line 8; page 12, line 23 - page 13, line 8, page 13, line 18 - page 14, line19; figures 2,3 and abstract --	1-24
X	US 6337881 B1 (CHADDHA, N.), 8 January 2002 (08.01.02), column 6, line 4 - line 24; column 9, line 50 - column 11, line 43, figures 2,4, abstract --	1-24
A	EP 1005232 A2 (GENERAL INSTRUMENT CORP), 31 May 2000 (31.05.00), [0075]-[0083], figures 1a-2b,3 and abstract --	1-24

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 April 2003	22 -04- 2003
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Alexander Lakic /LR Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1998)

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5774593 A (ZICK,G.L. ET AL), 30 June 1998 (30.06.98), figures 3A-5 --	1-24
A	US 6307886 B1 (WESTERMANN, E.F.), 23 October 2001 (23.10.01), column 2, line 28 - line 48; column 5, line 18 - line 30; column 5, line 63 - column 6, line 54, column 7, line 11 - line 28; figures 2,5, abstract -- --------	1-24

INTERNATIONAL SEARCH REPORT

Information on patent family members

28/02/03

International application No.
PCT/FI 03/00053

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 0147283 A1	28/06/01	AU 4338800 A	03/07/01
		EP 1249131 A	16/10/02
		TW 465243 B	00/00/00
US 6337881 B1	08/01/02	US 6233017 B	15/05/01
EP 1005232 A2	31/05/00	CN 1255022 A	31/05/00
		KR 2000035651 A	26/06/00
		TW 450006 B	00/00/00
		US 2002159523 A	31/10/02
US 5774593 A	30/06/98	NONE	
US 6307886 B1	23/10/01	JP 11275585 A	08/10/99
		SG 74691 A	22/08/00

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 July 2003 (31.07.2003)

PCT

(10) International Publication Number
WO 03/063505 A1

(51) International Patent Classification[7]: H04N 7/50

(21) International Application Number: PCT/FI03/00053

(22) International Filing Date: 22 January 2003 (22.01.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
20020127 23 January 2002 (23.01.2002) FI
10/306,942 29 November 2002 (29.11.2002) US

(71) Applicant *(for all designated States except US)*: **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **HANNUKSELA, Miska** [FI/FI]; Kukkaniitynkatu 4 B, FIN-33710 Tampere (FI).

(74) Agent: **KOLSTER OY AB**; Iso Roobertinkatu 23, P.O. Box 148, FIN-00121 Helsinki (FI).

(81) Designated States *(national)*: AE, AG, AL, AM, AT (utility model), AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (utility model), CZ, DE (utility model), DE, DK (utility model), DK, DM, DZ, EC, EE (utility model), EE, ES, FI (utility model), FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK (utility model), SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

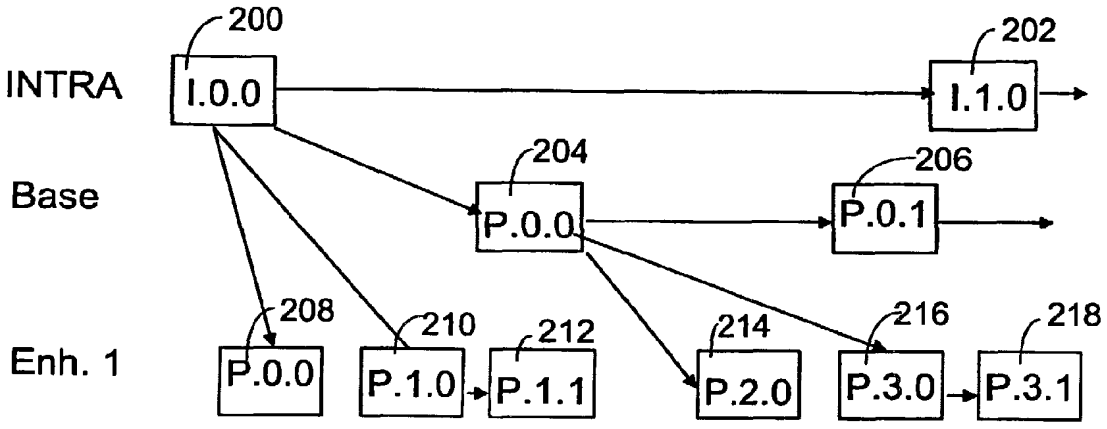
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for the following designations AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW, ARIPO patent (GH, GM, KE, LS,*

[Continued on next page]

(54) **Title:** GROUPING OF IMAGE FRAMES IN VIDEO CODING



(57) **Abstract:** A method for coding video frames for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format. The video frames of the first frame format are independent video frames, and the video frames of the second frame format are predicted from at least one of the other video frames. The video sequence has a first sub-sequence determined therein, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format; and with at least a second sub-sequence, at least part of which is formed by coding video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence. Frame identifier data of the second sub-sequence is determined into the video sequence.

WO 03/063505 A1

MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)

Published:

— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

GROUPING OF IMAGE FRAMES IN VIDEO CODING

FIELD OF THE INVENTION

[0001] The invention relates to the grouping of multimedia files, particularly video files and particularly in connection with streaming.

BACKGROUND OF THE INVENTION

[0002] The term 'streaming' refers to simultaneous sending and playback of data, typically multimedia data, such as audio and video files, in which the recipient may begin data playback already before all the data to be transmitted has been received. Multimedia data streaming systems comprise a streaming server and terminal devices that the recipients use for setting up a data connection, typically via a telecommunications network, to the streaming server. From the streaming server the recipients retrieve either stored or real-time multimedia data, and the playback of the multimedia data can then begin, most advantageously almost in real-time with the transmission of the data, by means of a streaming application included in the terminal.

[0003] From the point of view of the streaming server, the streaming may be carried out either as normal streaming or as progressive downloading to the terminal. In normal streaming the transmission of the multimedia data and/or the data contents are controlled either by making sure that the bit rate of the transmission substantially corresponds to the playback rate of the terminal device, or, if the telecommunications network used in the transmission causes a bottleneck in data transfer, by making sure that the bit rate of the transmission substantially corresponds to the bandwidth available in the telecommunications network. In progressive downloading the transmission of the multimedia data and/or the data contents do not necessarily have to be interfered with at all, but the multimedia files are transmitted as such to the recipient, typically by using transfer protocol flow control. The terminals then receive, store and reproduce an exact copy of the data transmitted from the server, which copy can then be later reproduced again on the terminal without needing to start a streaming again via the telecommunications network. The multimedia files stored in the terminal are, however, typically very large and their transfer to the terminal is time-consuming, and they require a significant amount of storage memory capacity, which is why a normal streaming is often preferred.

[0004] The video files in multimedia files comprise a great number of still image frames, which are displayed rapidly in succession (of typically 15 to 30 frames per s) to create an impression of a moving image. The image frames typically comprise a number of stationary background objects, determined by image information which remains substantially unchanged, and few moving objects, determined by image information that changes to some extent. The information comprised by consecutively displayed image frames is typically largely similar, i.e. successive image frames comprise a considerable amount of redundancy. The redundancy appearing in video files can be divided into spatial, temporal and spectral redundancy. Spatial redundancy refers to the mutual correlation of adjacent image pixels, temporal redundancy refers to the changes taking place in specific image objects in subsequent frames, and spectral redundancy to the correlation of different colour components within an image frame.

[0005] To reduce the amount of data in video files, the image data can be compressed into a smaller form by reducing the amount of redundant information in the image frames. In addition, while encoding, most of the currently used video encoders downgrade image quality in image frame sections that are less important in the video information. Further, many video coding methods allow redundancy in a bit stream coded from image data to be reduced by efficient, lossless coding of compression parameters known as VLC (Variable Length Coding).

[0006] In addition, many video coding methods make use of the above-described temporal redundancy of successive image frames. In that case a method known as motion-compensated temporal prediction is used, i.e. the contents of some (typically most) of the image frames in a video sequence are predicted from other frames in the sequence by tracking changes in specific objects or areas in successive image frames. A video sequence always comprises some compressed image frames the image information of which has not been determined using motion-compensated temporal prediction. Such frames are called INTRA-frames, or I-frames. Correspondingly, motion-compensated video sequence image frames predicted from previous image frames, are called INTER-frames, or P-frames (Predicted). The image information of P-frames is determined using one I-frame and possibly one or more previously coded P-frames. If a frame is lost, frames dependent on it can no longer be correctly decoded.

[0007] An I-frame typically initiates a video sequence defined as a Group of Pictures (GOP), the P-frames of which can only be determined on the basis of the I-frame and the previous P-frames of the GOP in question. The next I-frame begins a new group of pictures GOP, the image information comprised by which cannot thus be determined on the basis of the frames of the previous GOP. In other words, groups of pictures are not temporally overlapping, and each group of picture can be decoded separately. In addition, many video compression methods employ bi-directionally predicted B-frames (Bi-directional), which are set between two anchor frames (I- and P-frames, or two P-frames) within a group of pictures GOP, the image information of a B-frame being predicted from both the previous anchor frame and the one succeeding the B-frame. B-frames therefore provide image information of higher quality than P-frames, but typically they are not used as anchor frames, and therefore their removal from the video sequence does not degrade the quality of subsequent images. However, nothing prevents B-frames from being used as anchor frames as well, only in that case they cannot be removed from the video sequence without deteriorating the quality of the frames dependent on them.

[0008] Each video frame may be divided into what are known as macroblocks that comprise the colour components (such as Y, U, V) of all pixels of a rectangular image area. More specifically, a macroblock consists of at least one block per colour component, the blocks each comprising colour values (such as Y, U or V) of one colour level in the image area concerned. The spatial resolution of the blocks may differ from that of the macroblocks, for example U- and V-components may be displayed using only half of the resolution of Y-component. Macroblocks can be further grouped into slices, for example, which are groups of macroblocks that are typically selected in the scanning order of the image. Temporal prediction is typically carried out in video coding methods block- or macroblock-specifically, instead of image-frame-specifically.

[0009] To allow for flexible streaming of video files, many video coding systems employ scalable coding in which some elements or element groups of a video sequence can be removed without affecting the reconstruction of other parts of the video sequence. Scalability is typically implemented by grouping the image frames into a number of hierarchical layers. The image frames coded into the image frames of the base layer

substantially comprise only the ones that are compulsory for the decoding of the video information at the receiving end. The base layer of each group of pictures GOP thus comprises one I-frame and a necessary number of P-frames. One or more enhancement layers can be determined below the base layer, each one of the layers improving the quality of the video coding in comparison with an upper layer. The enhancement layers thus comprise P- or B-frames predicted on the basis of motion-compensation from one or more upper layer images. The frames are typically numbered according to an arithmetical series.

[0010] In streaming, transmission bit rate must be controllable either on the basis of the bandwidth to be used or the maximum decoding or bit rate value of the recipient. Bit rate can be controlled either at the streaming server or in some element of the telecommunications network, such as an Internet router or a base station of a mobile communications network. The simplest means for the streaming server to control the bit rate is to leave out B-frames having a high information content from the transmission. Further, the streaming server may determine the number of scalability layers to be transmitted in a video stream, and thus the number of the scalability layers can be changed always when a new group of pictures GOP begins. It is also possible to use different video sequence coding methods. Correspondingly, B-frames, as well as other P-frames of the enhancement layers, can be removed from the bit stream in a telecommunications network element.

[0011] The above arrangement involves a number of drawbacks. Many coding methods, such as the coding according to the ITU-T (International Telecommunications Union, Telecommunications Standardization Sector) standard H.263, are familiar with a procedure called reference picture selection. In reference picture selection at least a part of a P-image has been predicted from at least one other image than the one immediately preceding the P-image in the time domain. The selected reference image is signalled in a coded bit stream or in bit stream header fields image-, image-segment- (such as a slice or a group of macroblocks), macroblock-, or block-specifically. The reference picture selection can be generalized such that the prediction can also be made from images temporally succeeding the image to be coded. Further, the reference picture selection can be generalized to cover all temporally predicted frame types, including B-frames. Since it is possible to also select at least one image preceding an I-image that begins a

group of pictures GOP as the reference image, a group of pictures employing reference picture selection cannot necessarily be decoded independently. In addition, the adjusting of scalability or coding method in the streaming server or a network element becomes difficult, because the video sequence must be decoded, parsed and buffered for a long period of time to allow any dependencies between different image groups to be detected.

[0012] Another problem in the prior art coding methods is that there is no useful method of signalling differences of significance between INTER-frames. For example, when a plural number of P-frames are to be predicted successively, the first P-frames are usually the most significant ones for the reconstruction because there are more image frames that are dependent on the first P-frame than on subsequent P-frames. However, known coding methods fail to provide a simple method for signalling such differences in significance.

[0013] A further problem relates to the insertion of a video sequence in the middle of another video sequence, which has typically led to discontinuity in image numbering. The numbering of video sequence images is typically used for detecting the loss of image frames. However, if a separate video sequence, such as a commercial, is inserted into a video sequence, the separate video sequence is typically provided with separate image numbering, which is not in line with the ascending image numbering of the original video sequence. The receiving terminal may therefore interpret the deviating image numbering as a signal of lost image frames and start unnecessary actions to reconstruct the image frames suspected as lost or to request a re-transmission thereof. A similar problem is encountered when scaling a video sequence: for example, if a plural number of successive frames is removed, the receiving terminal may unnecessarily interpret these removals as protocol errors.

BRIEF DESCRIPTION OF THE INVENTION

[0014] It is therefore an object of the invention to provide a method and equipment implementing the method that allow the disadvantages caused by the above problems to be reduced. The objects of the invention are achieved by a method, video encoder, video decoder, streaming system element and computer programs that are chacterized by what is stated in the independent claims.

[0015] The preferred embodiments of the invention are disclosed in the dependent claims.

[0016] The invention is based on the idea of coding a scalable, compressed video sequence that comprises video frames coded on the basis of at least a first and a second frame format, the video frames conforming to the first frame format being independent of other video frames, i.e. they are typically I-frames, and the video frames according to the second frame format being frames predicted from at least one other video frame, for example P-frames. A video sequence comprises a first sub-sequence formed therein, at least part of the sub-sequence being formed by coding video frames (I-frames) of the at least first frame format (I-frames), and at least a second sub-sequence, at least part of which is formed by coding video frames of the at least second frame format (e.g. P-frames), at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence. In addition, identifying data of the video frames comprised by the at least second sub-sequence are determined into the video sequence in question.

[0017] Consequently, an essential aspect of the present invention is to determine the sub-sequences each sub-sequence is dependent on, i.e. a sub-sequence will comprise information of all sub-sequences that have been directly used for predicting the image frames comprised by the sub-sequence in question. This information is signalled in the bit stream of the video sequence, preferably separate from the actual image information, whereby the image data comprised by the video sequence can preferably be scaled, because independently decodable portions of the video sequence can be easily determined, and the portions can be removed without affecting the decoding of the rest of the image data.

[0018] According to a preferred embodiment of the invention, a scalable coding hierarchy is formed for the video sequences, according to which hierarchy a first scalability layer of the video sequence is coded in such a way that it comprises at least video frames of the first frame format, i.e. I-frames, and the lower scalability layers of the video sequence are coded such that they comprise at least video frames of the second frame format, i.e. P- and/or B-frames, grouped into sub-sequences in which at least one video frame is predicted either from a video frame of an upper scalability layer or

from another video frame of the same sub-sequence. The number of the scalability layers is not restricted.

[0019] According to a preferred embodiment of the invention, a unique identifier combining the scalability layer number, sub-sequence identifier and image number is determined for each video frame. The identifier can be included into a header field of the video sequence, or into a header field according to the transfer protocol to be used for the video sequence transfer.

[0020] An advantage of the procedure of the invention is that it provides a flexible coding hierarchy that enables, on one hand, the scaling of the bit rate of a video sequence to be transmitted without video sequence decoding being required, and, on the other hand, independent decoding of each sub-sequence. This allows the streaming server, for example, to adjust the bit rate advantageously, without video sequence decoding, parsing and buffering, because the streaming server is capable of deducing the inter-dependencies of the different sub-sequences directly from the identifiers and their inter-dependencies. Further, the streaming server may, when necessary, carry out sub-sequence-specific adjusting of the scalability or the coding employed, because the inter-dependencies of the different frames are known. A further advantage is that the coding hierarchy and the image numbering of the invention allow a separate video sequence to the easily inserted into another video sequence.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] In the following, the invention will be described in connection with the preferred embodiments and with reference to the accompanying drawings, in which

Figure 1 illustrates a common multimedia data streaming system in which the scalable coding hierarchy of the invention can be applied;

Figure 2 illustrates a scalable coding hierarchy of a preferred embodiment of the invention;

Figures 3a and 3b illustrate embodiments of the invention for adjusting scalability;

Figures 4a, 4b and 4c illustrate embodiments of the invention for adjusting image numbering;

Figures 5a, 5b and 5c illustrate embodiments of the invention for using B-frames in a scalable coding hierarchy;

Figures 6a, 6b and 6c illustrate scalable coding hierarchies of preferred embodiments of the invention in connection with reference picture selection; and

Figure 7 illustrates an arrangement according to a preferred embodiment of the invention for coding scene transition.

DETAILED DESCRIPTION OF THE INVENTION

[0022] In the following, a general-purpose multimedia data streaming system is disclosed, the basic principles of which can be applied in connection with any telecommunications system. Although the invention is described here with a particular reference to a streaming system, in which the multimedia data is transmitted most preferably through a telecommunications network employing a packet-switched data protocol, such as an IP network, the invention can equally well be implemented in circuit-switched networks, such as fixed telephone networks PSTN/ISDN (Public Switched Telephone Network/Integrated Services Digital Network) or in mobile communications networks PLMN (Public Land Mobile Network). Further, the invention can be applied in the streaming of multimedia files in the form of both normal streaming and progressive downloading, and for implementing video calls, for example.

[0023] It is also to be noted that although the invention is described here with a particular reference to streaming systems and the invention can also be advantageously applied in them, the invention is not restricted to streaming systems alone, but can be applied in any video reproduction system, irrespective of how a video file that is to be decoded is downloaded and where it is downloaded from. The invention can therefore be applied for example in the playback of a video file to be downloaded from a DVD disc or from some other computer memory carrier, for example in connection with varying processing capacity available for video playback. In particular, the invention can be applied to different video codings of low bit rate that are typically used in telecommunications systems subject to bandwidth restrictions. One example is the system defined in the ITU-T standard H.263 and the one that is being defined in H.26L (possibly later to become H.264). In connection with these, the invention can be applied to mobile stations, for example, in which case the

video playback can be made to adjust both to changing transfer capacity or channel quality and to the processor power currently available, when the mobile station is used also for executing other applications than video playback.

[0024] It is further to be noted that, for the sake of clarity, the invention will be described below by giving an account of image frame coding and temporal predicting on image frame level. However, in practice coding and temporal predicting typically take place on block or macroblock level, as described above.

[0025] With reference to Figure 1, a typical multimedia streaming system will be described, which is a preferred system for applying the procedure of the invention.

[0026] A multimedia data streaming system typically comprises one or more multimedia sources 100, such as a video camera and a microphone, or video image or computer graphic files stored in a memory carrier. Raw data obtained from the different multimedia sources 100 is combined into a multimedia file in an encoder 102, which can also be referred to as an editing unit. The raw data arriving from the one or more multimedia sources 100 is first captured using capturing means 104 included in the encoder 102, which capturing means can be typically implemented as different interface cards, driver software, or application software controlling the function of a card. For example, video data may be captured using a video capture card and the associated software. The output of the capturing means 104 is typically either an uncompressed or slightly compressed data flow, for example uncompressed video frames of the YUV 4:2:0 format or motion-JPEG image format, when a video capture card is concerned.

[0027] An editor 106 links different media flows together to synchronize video and audio flows to be reproduced simultaneously as desired. The editor 106 may also edit each media flow, such as a video flow, by halving the frame rate or by reducing spatial resolution, for example. The separate, although synchronized, media flows are compressed in a compressor 108, where each media flow is separately compressed using a compressor suitable for the media flow. For example, video frames of the YUV 4:2:0 format may be compressed using the low bit rate video coding according to the ITU-T recommendation H.263 or H.26L. The separate, synchronized and compressed media flows are typically interleaved in a multiplexer 110, the

output obtained from the encoder 102 being a single, uniform bit flow that comprises data of a plural number of media flows and that may be referred to as a multimedia file. It is to be noted that the forming of a multimedia file does not necessarily require the multiplexing of a plural number of media flows into a single file, but the streaming server may interleave the media flows just before transmitting them.

[0028] The multimedia files are transferred to a streaming server 112, which is thus capable of carrying out the streaming either as real-time streaming or in the form of progressive downloading. In progressive downloading the multimedia files are first stored in the memory of the server 112 from where they may be retrieved for transmission as need arises. In real-time streaming the editor 102 transmits a continuous media flow of multimedia files to the streaming server 112, and the server 112 forwards the flow directly to a client 114. As a further option, real-time streaming may also be carried out such that the multimedia files are stored in a storage that is accessible from the server 112, from where real-time streaming can be driven and a continuous media flow of multimedia files is started as need arises. In such case, the editor 102 does not necessarily control the streaming by any means. The streaming server 112 carries out traffic shaping of the multimedia data as regards the bandwidth available or the maximum decoding and playback rate of the client 114, the streaming server being able to adjust the bit rate of the media flow for example by leaving out B-frames from the transmission or by adjusting the number of the scalability layers. Further, the streaming server 112 may modify the header fields of a multiplexed media flow to reduce their size and encapsulate the multimedia data into data packets that are suitable for transmission in the telecommunications network employed. The client 114 may typically adjust, at least to some extent, the operation of the server 112 by using a suitable control protocol. The client 114 is capable of controlling the server 112 at least in such a way that a desired multimedia file can be selected for transmission to the client, in addition to which the client is typically capable of stopping and interrupting the transmission of a multimedia file.

[0029] When the client 114 is receiving a multimedia file, the file is first supplied to a demultiplexer 116, which separates the media flows comprised by the multimedia file. The separate, compressed media flows are then supplied to a decompressor 118 where each separate media flow is decompressed by a decompressor suitable for each particular media flow. The

decompressed and reconstructed media flows are supplied to a playback unit 120 where the media flows are rendered at a correct pace according to their synchronization data and supplied to presentation means 124. The actual presentation means 124 may comprise for example a computer or mobile station display, and loudspeaker means. The client 114 also typically comprises a control unit 122 that the end user can typically control through a user interface and that controls both the operation of the server, through the above-described control protocol, and the operation of the playback unit 120, on the basis of the instructions given by the end user.

[0030] It is to be noted that the transfer of multimedia files from the streaming server 112 to the client 114 takes place through a telecommunications network, the transfer path typically comprising a plural number of telecommunications network elements. It is therefore possible that there is at least some network element that can carry out traffic shaping of multimedia data with regard to the available bandwidth or the maximum decoding and playback rate of the client 114 at least partly in the same way as described above in connection with the streaming server.

[0031] Scalable coding will be described below with reference to a preferred embodiment of the invention and an example illustrated in Figure 2. Figure 2 shows part of a compressed video sequence having a first frame 200, which is an INTRA frame, or I-frame, and thus an independently determined video frame the image information of which is determined without using motion-compensated temporal prediction. The I-frame 200 is placed on a first scalability layer, which may be referred to as an INTRA layer. Each scalability layer is assigned a unique identifier, such as the layer number. The INTRA layer may therefore be given the number 0, for example, or some other alphanumeric identifier, for example a letter, or a combination of a letter and a number.

[0032] Correspondingly, sub-sequences consisting of groups of one or more video frames are determined for each scalability layer, at least one of the images in a group (typically the first or the last one) being temporally predicted at least from a video frame of another, sub-sequence of typically either a higher or the same scalability layer, the rest of the video frames being temporally predicted either from only the video frames of the same sub-sequence, or possibly also from one or more video frames of said second sub-sequence. A sub-sequence may de decoded independently, irrespective of

other sub-sequences, except said second sub-sequence. The sub-sequences of each scalability layer are assigned a unique identifier using for example consecutive numbering starting with number 0 given for the first sub-sequence of a scalability layer. Since the I-frame 200 is determined independently and can also be decoded independently upon reception, irrespective of other image frames, it also forms in a way a separate sub-sequence.

[0033] An essential aspect of the present invention is therefore to determine each sub-sequence in terms of those sub-sequences the sub-sequence is dependent on. In other words, a sub-sequence comprises information about all the sub-sequences that have been directly used for predicting the image frames of the sub-sequence in question. This information is signalled in a video sequence bit stream, preferably separate from the actual image information, and therefore the image data of the video sequence can be preferably adjusted because it is easy to determine video sequence portions that are to be independently decoded and can be removed without affecting the decoding of the rest of the image data.

[0034] Next, within each sub-sequence, the video frames of the sub-sequence are given image numbers, using for example consecutive numbering that starts with the number 0 given to the first video frame of the sub-sequence. Since the I-frame 200 also forms a separate sub-sequence, its image number is 0. In Figure 2, the I-frame 200 shows the type (I), sub-sequence identifier and image number (0.0) of the frame.

[0035] Figure 2 further shows a next I-frame 202 of the INTRA layer, the frame thus being also an independently determined video frame that has been determined without using motion-compensated temporal prediction. The temporal transmission frequency of I-frames depends on many factors relating to video coding, image information contents and the bandwidth to be used, and, depending on the application or application environment, I-frames are transmitted in a video sequence at intervals of 0.5 to 10 seconds, for example. Since the I-frame 202 can be independently decoded, it also forms a separate sub-sequence. Since this is the second sub-sequence in the INTRA-layer, the consecutive numbering of the sub-sequence identifier of the I-frame 202 is 1. Further, since the I-frame 202 also forms a separate sub-sequence, i.e. it is the only video frame in the sub-sequence, its image number is 0. The I-frame 202 can thus be designated with identifier (I.1.0.) Correspondingly, the identifier of the next I-frame on the INTRA layer is (I.2.0.), etc. As a result, only

independently determined I-frames in which the image information is not determined using motion-compensated temporal prediction are coded into the first scalability layer, i.e. the INTRA layer. The sub-sequences can also be determined using other kind of numbering or other identifiers, provided that the sub-sequences can be distinguished from one another.

[0036] The next scalability layer, which has a layer number 1, for example, and which may be referred to as the base layer, comprises coded, motion-compensated INTER or P-frames typically predicted only from previous image frames, i.e. in this case from the I-frames of an upper INTRA layer. The image information of the first P-frame 204 of the base layer shown in Figure 2 is determined using the I-frame 200 of the INTRA layer. A P-frame 204 begins the first sub-sequence of the base layer, and therefore the sub-sequence identifier of the P-frame 204 is 0. Further, since the P-frame 204 is the first image frame of the first sub-sequence of the base layer, the image number of the P-frame 204 is 0. The P-frame 204 can thus be identified with (P.0.0).

[0037] The temporally succeeding P-frame 206 of the base layer is predicted from the previous P-frame 204. The P-frames 204 and 206 thus belong to the same sub-sequence, whereby the P-frame 206 also receives the sub-sequence identifier 0. Since the P-frame 206 is the second image frame in the sub-sequence 0, the image number of the P-frame 206 is 1, and the P-frame 206 can be identified with (P.0.1).

[0038] The scalability layer succeeding the base layer and having the layer number 2, is called enhancement layer 1. This layer comprises coded, motion-compensated P-frames predicted only from previous image frames, in this case either from I-frames of the INTRA layer or P-frames of the base layer. Figure 2 shows a first image frame 208 and a second image frame 210 of enhancement layer 1, which are both predicted only from the first image frame 200 of the INTRA layer. The P-frame 208 begins the first sub-sequence of enhancement layer 1, the sub-sequence identifier of the P-frame thus being 0. Further, since the P-frame 208 is the first and only image frame in said sub-sequence, the P-frame 208 receives the image number 0. The P-frame 208 can thus be identified with (P.0.0.).

[0039] Since the second image frame 210 is also predicted only from the first image frame 200 of the INTRA layer, the P-frame 210 begins the second sub-sequence of enhancement layer 1 and the sub-sequence identifier of the P-frame 210 is therefore 1. Since the P-frame 210 is the first image

frame in the sub-sequence, the image number of the P-frame 210 is 0. The P-frame can thus be identified with (P.1.0.). The temporally succeeding P-frame 212 of enhancement layer 1 is predicted from the previous P-frame 210. The P-frames 210 and 212 thus belong to the same sub-sequence, and therefore the P-frame also receives the sub-sequence identifier 1. The P-frame 212 is the second image frame in the sub-sequence 1, and therefore the P-frame receives the image number 1 and can be identified with (P.1.1).

[0040] The temporally fourth image frame 214 of enhancement layer 1 is predicted from the first image frame 204 of the base layer. The P-frame 214 thus begins a third sub-sequence of enhancement layer 1 and therefore the P-frame 214 receives the sub-sequence identifier 2. Further, since the P-frame 214 is the first and only image frame in the sub-sequence, the image number of the P-frame 214 is 0. The P-frame 208 can therefore be identified with (P.2.0).

[0041] Also the temporally fifth image frame 216 of enhancement layer 1 is predicted only from the first image frame 204 of the base layer, the P-frame 216 thus beginning the fourth sub-sequence of enhancement layer 1, and the sub-sequence identifier of the P-frame 216 is 3. In addition, since the P-frame 216 is the first one in the sub-sequence in question, the image number of the P-frame 216 is 0. The P-frame 216 can therefore be identified with (P.3.0). The temporally following P-frame 218 of enhancement layer 1 is predicted from the previous P-frame 216. The P-frames 216 and 218 thus belong to the same sub-sequence, and the sub-sequence identifier of the P-frame 218 is also 3. Since the P-frame 218 is the second image frame in the sub-sequence 3, the image number of the P-frame 218 is 1 and the identifier of the P-frame 218 is (P.3.1).

[0042] For simplicity and clarity of illustration the above disclosure only relates to I- and P-frames. However, a person skilled in the art will find it apparent that the scalable video coding of the invention can also be implemented using other known image frame types, such as the above described B-frames and at least SI-frames, SP-frames and MH-frames. SI-frames correspond to I-frames, but together with an SP-frame they allow an identical image to be reconstructed. An SP-frame, in turn, is a P-frame subjected to a particular coding that allows an identical image to be reconstructed together with an SI-frame or another SP-frame. SP-frames are typically placed into a video sequence into points where an access point or a

scanning point is desired, or where the changing of the coding parameters of the video stream should be possible. The frames can also be used for error correction and for increasing error tolerance. SP-frames are otherwise similar to ordinary P-frames predicted from previous frames, except that they are defined so that they can be replaced by another video frame of the SP- or SI-type, the result of the decoding of the new frame being identical with the decoding result of the original SP-frame that was in the video stream. In other words, a new SP-frame that is used for replacing the one that was in the video stream is predicted from another sequence or video stream and yet the reconstructed frame has identical contents. SP-frames are described for example in the Applicant's earlier application PCT/FI02/00004.

[0043] Similarly as B-frames, macroblocks of MH (Multi Hypothesis) frames, based on motion-compensated prediction, are predicted from two other frames, which are not, however, necessarily located next to an MH-frame. More precisely, the predicted macroblocks are computed as an average of two macroblocks of two other frames. Instead of two frames, MH-frame macroblocks can naturally be also predicted from one other frame. Reference images may change according to macroblock, in other words, all macroblocks in one and the same image are not necessarily predicted using the same frames.

[0044] A sub-sequence thus covers a specific period of time in a video sequence. The sub-sequences of the same layer or of different layers may be partly or entirely overlapping. If there are temporally overlapping image frames on the same layer, the frames are interpreted as alternative presentations of the same image content and therefore any mode of image presentation can be used. On the other hand, if there are temporally overlapping image frames on different layers, they form different presentations of the same image content, and therefore presentations differ in image quality, i.e. the quality of image is better on a lower layer.

[0045] The above disclosure referring to Figure 2 illustrates a scalable coding arrangement and a hierarchical structure and numbering of image frames according to a preferred embodiment of the invention. In this embodiment the INTRA-layer only comprises I-frames and the base layer can only be decoded using the information received from the INTRA-layer. Correspondingly, the decoding of enhancement layer 1 typically requires information from both the base layer and the INTRA-layer.

[0046] The number of the scalability layers is not restricted to three, as above, but any number of enhancement layers that is considered necessary for producing sufficient scalability may be used. Consequently, the layer number of enhancement layer 2 is four, that of enhancement layer 3 is five, etc. Since some of the image frames in the above example are given the same identifier (e.g. the identifier of both image frames 204 and 208 is (P.0.0)), by including the layer number in the identifier each image frame can be uniquely identified and, at the same time, the dependencies of each image frame on other image frames is preferably determined. Each image frame is thus uniquely identified, the identifier of image frame 204, for example, being (P.1.0.0), or simply (1.0.0) and, correspondingly, that of image 208 being (P.2.0.0), or (2.0.0).

[0047] According to a preferred embodiment of the invention, the number of a reference image frame is determined according to a specific, predetermined alpha-numeric series, for example as an integer between 0 and 255. When the parameter value achieves the maximum value N (e.g. 255) in the series concerned, the determining of the parameter value starts from the beginning, i.e. from the minimum value of the series (e.g. 0). An image frame is thus uniquely identified within a specific sub-sequence up to the point where the same image number is used again. The sub-sequence identifier can also be determined according to a specific, predetermined arithmetic series. When the value of the sub-sequence identifier achieves the maximum value N of the series, the determining of the identifier starts again from the beginning of the series. However, a sub-sequence cannot be assigned an identifier that is still in use (within the same layer). The series in use may also be determined in another way than arithmetically. One alternative is to assign random sub-sequence identifiers, taking into account that an assigned identifier is not be used again.

[0048] A problem in the numbering of image frames arises when the user wishes to start browsing a video file in the middle of a video sequence. Such situations occur for example when the user wishes to browse a locally stored video file backward or forward or to browse a streaming file at a particular point; when the user initiates the playback of a streaming file from a random point; or when a video file that is to be reproduced is detected to contain an error that interrupts the playback or requires the playback to be resumed from a point following the error. When the browsing of a video file is

resumed from a random point after previous browsing, discontinuity typically occurs in the image numbering. The decoder typically interprets this as unintentional loss of image frames and unnecessarily tries to reconstruct the image frames suspected as lost.

[0049] According to a preferred embodiment of the invention, this can be avoided in the decoder by defining an initiation image in an independently decodable Group of Pictures GOP that is activated at the random point of the video file, and the number of the initiation image is set at zero. This independently decodable image group can thus be a sub-sequence of an INTRA-layer, for example, in which case an I-frame is used as the initiation image, or, if scaling originating from the base layer is employed, the independently decodable image group is a sub-sequence of the base layer, in which case the first image frame of the sub-sequence, typically an I-frame, is usually used as the initiation image. Consequently, when activated at a random point, the decoder preferably sets the identifier of the first image frame, preferably an I-frame, of the independently decodable sub-sequence at zero. Since the sub-sequence to be decoded may also comprise other image frames whose identifier is zero (for example when the above described alpha-numeric series starts from the beginning), the beginning of the sub-sequence, i.e. its first image frame, can be indicated to the decoder for example by a separate flag added to the header field of a slice of the image frame. This allows the decoder to interpret the image numbers correctly and to find the correct image frame that initiates the sub-sequence from the video sequence image frames.

[0050] The above numbering system provides only one example of how the unique image frame identification of the invention can be carried out so that interdependencies between the image frames are indicated at the same time. However, video coding methods in which the method of the invention can be applied, such as video coding methods according to the ITU-T standards H.263 and H.26L, employ code tables, which in turn use variable length codes. When variable length codes are used for coding layer numbers, for example, a lower code word index, i.e. a smaller layer number, signifies a shorter code word. In practice the scalable coding of the invention will be used in most cases in such a way that the base layer will consist significantly more image frames than the INTRA-layer. This justifies the use of a lower index, i.e. a smaller layer number, on the base layer than on the INTRA-layer, because

the amount of coded video data is thereby advantageously reduced. Consequently, the INTRA-layer is preferably assigned layer number 1 and the base layer is given layer number 0. Alternatively, the code can be formed by using fewer bits for coding the base layer number than the INTRA-layer number, in which case the actual layer number value is not relevant in view of the length of the code created.

[0051] Further, according to a second preferred embodiment of the invention, when the number of the scalability layers is to be kept low, the first scalability layer in particular can be coded to comprise both the INTRA-layer and the base layer. From the point of view of coding hierarchy, the simplest way to conceive this is to leave out the INTRA-layer altogether, and to provide the base layer with coded frames consisting of both independently defined I-frames, the image information of which has not been determined using motion-compensated temporal prediction, and image frames predicted from previous frames, which image frames in this case are motion-compensated P-frames predicted from the I-frames of the same layer. The layer number 0 can thus still be used for the base layer and, if enhancement layers are coded into the video sequence, enhancement layer 1 is assigned layer number 1. This is illustrated in the following, with reference to Figures 3a and 3b.

[0052] Figure 3a shows a non-scalable video sequence structure, in which all image frames are placed on the same scalability layer, i.e. the base layer. The video sequence comprises a first image frame 300 which is an I-frame (I.0.0) and which thus initiates a first sub-sequence. The image frame 300 is used for predicting a second image frame 302 of the sub-sequence, i.e. a P-frame (P.0.1), which is then used for predicting a third image frame 304 of the sub-sequence, i.e. a P-frame (P.0.2), which is in turn used for predicting the next image frame 306, i.e. a P-frame (P.0.3). The video sequence is then provided with an I-frame (I.1.0) coded therein, i.e. an I-frame 308, which thus initiates a second sub-sequence in the video sequence. This kind of non-scalable coding can be used for example when the application employed does not allow scalable coding to be used, or there is no need for it. In a circuit-switched videophone application, for example, channel bandwidth remains constant and the video sequence is coded in real-time, and therefore there is typically no need for scalable coding.

[0053] Figure 3b, in turn, illustrates an example of how scalability can be added, when necessary, to a combined INTRA- and base layer. Here,

too, the video sequence base layer comprises a first image frame 310 which is an I-frame (I.0.0) and which initiates a first sub-sequence of the base layer. The image frame 310 is used for predicting a second image frame 312 of the sub-sequence, i.e. a P-frame (P.0.1), which is then used for predicting a third image frame 314 of the sub-sequence, i.e. a P-frame (P.0.2). Enhancement layer 1, however, is also coded into this video sequence and it comprises a first sub-sequence, the first and only image frame 316 of which is a P-frame (P.0.0), which is predicted from the first image frame 310 of the base layer. The first image frame 318 of a second sub-sequence of the enhancement layer is, in turn, predicted from the second image frame 312 of the base layer, and therefore the identifier of this P-frame is (P.1.0). The next image frame 320 of the enhancement layer is again predicted from the previous image frame 318 of the same layer and, therefore, it belongs to the same sub-sequence, its identifier thus being (P.1.1).

[0054] In this embodiment of the invention the sub-sequences of the base layer can be decoded independently, although a base layer sub-sequence may be dependent on another base layer sub-sequence. The decoding of the base layer sub-sequences requires information from the base layer and/or from the second sub-sequence of enhancement layer 1, the decoding of the sub-sequences of enhancement layer 2 requires information from enhancement layer 1 and/or from the second sub-sequence of enhancement layer 2, etc. According to an embodiment, I-frames are not restricted to the base layer alone, but lower enhancement layers may also comprise I-frames.

[0055] The basic idea behind the above embodiments is that a sub-sequence comprises information about all the sub-sequences it is dependent on, i.e. about all sub-sequences that have been used for predicting at least one of the image frames of the sub-sequence in question. However, according to an embodiment it is also possible that a sub-sequence comprises information about all sub-sequences that are dependent on the sub-sequence in question, in other words, about all the sub-sequences in which at least one image frame has been predicted using at least one image frame of the sub-sequence in question. Since in the latter case the dependencies are typically determined temporally forward, image frame buffers can be advantageously utilized in the coding in a manner to be described later.

[0056] In all the above embodiments the numbering of the image frames is sub-sequence-specific, i.e. a new sub-sequence always starts the numbering from the beginning. The identification of an individual image frame thus requires the layer number, sub-sequence identifier and image frame number to be determined. According to a preferred embodiment of the invention, the image frames can be independently numbered using consecutive numbering in which successive reference image frames in the coding order are indicated with numbers incremented by one. As regards layer numbers and sub-sequence identifiers, the above-described numbering procedure can be used. This allows each image frame to be uniquely identified, when necessary, without using the layer number and sub-sequence identifier.

[0057] This is illustrated with the example shown in Figure 4a in which the base layer comprises a temporally first I-frame 400 (I.0.0). This frame is used for predicting a first image frame 402 of enhancement layer 1, i.e. (P.0.1), which is then used for predicting a second image frame 404 belonging to the same sub-sequence (with sub-sequence identifier 0), i.e. (P.0.2), which is used for predicting a third image frame 406 of the same sub-sequence, i.e. (P.0.3), which is used for predicting a fourth image frame 408 (P.0.4) and, finally, the fourth frame for predicting a fifth image frame 410 (P.0.5). The temporally next video sequence image frame 412 is located on the base layer, where it is in the same sub-sequence as the I-frame 400, although temporally it is only the seventh coded image frame, and therefore its identifier is (P.0.6). The seventh frame is then used for predicting a first image frame 414 of the second sub-sequence of enhancement layer 1, i.e. (P.1.7), which is then used for predicting a second image frame 416 belonging to the same sub-sequence (with sub-sequence identifier 1), i.e. (P.1.8), which in turn used for predicting a third image frame 418 (P.1.9), the third for predicting a fourth image frame 420 (P.1.10) and, finally, the fourth for predicting a fifth image frame 422 (P.1.11) of the same sub-sequence. Again, the temporally next video sequence image frame 424 is located on the base layer, where it is in the same sub-sequence as the I-frame 400 and the P-frame 412, although temporally it is only the thirteenth coded image frame and therefore its identifier is (P.0.12). For clarity of illustration, the above description of the embodiment does not comprise layer identifiers, but it is apparent that in order

to implement scalability, also the layer identifier must be signalled together with the video sequence, typically as part of the image frame identifiers.

[0058] Figures 4b and 4c show alternative embodiments for grouping the image frames of the video sequence shown in Figure 4a. The image frames in Figure 4b are numbered according to sub-sequence, i.e. a new sub-sequence always starts the numbering from the beginning (from zero). Figure 4c, in turn, employs image frame numbering which corresponds otherwise to that used in Figure 4a, except that the P-frames of the base layer are replaced by SP-frame pairs to allow for identical reconstruction of image information.

[0059] As stated above, the procedure of the invention can also be implemented using B-frames. One example of this is illustrated in Figures 5a, 5b and 5c. Figure 5a shows a video sequence in the time domain, the sequence comprising P-frames P1, P4 and P7, with B-frames placed between them, the interdependencies of the B-frames with regard to temporal predicting being shown with arrows. Figure 5b shows a preferred grouping of video sequence image frames in which the interdependencies shown in Figure 5a are indicated. Figure 5b illustrates sub-sequence-specific image frame numbering in which a new sub-sequence always starts the numbering of the image frames from zero. Figure 5c, in turn, illustrates image frame numbering, which is consecutive in the order of temporal prediction, wherein the following reference frame always receives the next image number as the previously encoded reference frame. The image frame (B1.8) (and (B2.10)) does not serve as a reference prediction frame to any other frame, therefore it does not affect the image frame numbering.

[0060] The above examples illustrate different alternatives of how scalability of video sequence coding can be adjusted by using the method of the invention. From the point of view of the terminal device reproducing the video sequence, the more scalability layers are available, or the more scalability layers it is capable of decoding, the better the image quality. In other words, increase in the amount of image information and in the bit rate used for transferring the information improves the temporal or spatial resolution, or the spatial quality of the image data. Correspondingly, a higher number of scalability layers also sets considerably higher demands on the processing capacity of the terminal device performing decoding.

[0061] In addition, the above examples illustrate the advantage gained by using sub-sequences. With image frame identifiers, the dependencies of each image frame from other image frames in the sub-sequence are indicated in an unambiguous manner. A sub-sequence thus forms an independent whole that can be left out of a video sequence, when necessary, without affecting the decoding of subsequent image frames of the video sequence. In that case, only the image frames of the sub-sequence in question and of those sub-sequences on the same and/or lower scalability layers dependent on it are not decoded.

[0062] The image frame identifier data transmitted together with the video sequence are preferably included in the video sequence header fields or in the header fields of the transfer protocol to be used for transferring the video sequence. In other words, the identifier data of predicted image frames are not included in the image data of the coded video sequence, but always into the header fields, whereby the dependencies of the image frames can be detected without decoding the images of the actual video sequence. The identifier data of the image frames can be stored for example in the buffer memory of the streaming server as the video sequence is being coded for transmission. In addition, the sub-sequences can be independently decoded on each scalability layer, because the image frames of a sub-sequence are not dependent on other sub-sequences of the same scalability layer.

[0063] According to an embodiment of the invention, the image frames comprised by a sub-sequence may thus depend also on other sub-sequences of the same scalability layer. This dependency must then be signalled for example to the streaming server carrying out traffic shaping, because interdependent sub-sequences located on the same layer cannot be separately removed from a video sequence to be transmitted. A preferred way to carry out the signalling is to include it in the image frame identifiers to be transmitted, for example by listing the layer-sub-sequence pairs the sub-sequence in question depends on. This also provides a preferred way of indicating dependency from another sub-sequence of the same scalability layer.

[0064] The above examples illustrate a situation where image frames are temporally predicted from previous image frames. In some coding methods, however, the reference picture selection has been further extended to also include the predicting of the image information of image frames from

temporally succeeding image frames. Reference picture selection offers most diversified means for creating different temporally scalable image frame structures and allows the error sensitivity of the video sequence to be reduced. One of the coding techniques based on reference picture selection is INTRA-frame postponement. The INTRA-frame is not placed into its temporally "correct" position in the video sequence, but its position is temporally postponed. The video sequence image frames that are between the "correct" position of the INTRA-frame and its actual position are predicted temporally backward from the INTRA-frame in question. This naturally requires that uncoded image frames be buffered for a sufficiently long period of time so that all image frames that are to be displayed can be coded and arranged into their order of presentation. INTRA-frame transfer and the associated determining of sub-sequences in accordance with the invention are illustrated in the following with reference to Figure 6.

[0065] Figure 6a shows a video sequence part in which the INTRA-frame comprises a single I-frame 600, which is temporally transferred to the position shown in Figure 6, although the "correct" position of the I-frame in the video sequence would have been at the first image frame. The video sequence image frames between the "correct" position and the real position 600 are thus temporally predicted backward from the I-frame 600. This is illustrated by a sub-sequence coded into enhancement layer 1 and having a first temporally backward predicted image frame 602, which is a P-frame (P.0.0). This frame is used for temporally predicting a previous image frame 604, i.e. a P-frame (P.0.1), which is used in turn for predicting an image frame 606, i.e. a P-frame (P.0.2), and, finally, the frame 606 for predicting an image frame 608, i.e. a P-frame (P.0.3), which is at the position that would have been the "correct" position of the I-frame 600 in the video sequence. Correspondingly, the I-frame 600 on the base layer is used for temporally forward prediction of a sub-sequence comprising four P-frames 610, 612, 614 and 616, i.e. P-frames (P.0.0), (P.0.1), (P.0.2) and (P.0.3).

[0066] The fact that in this example backward predicted image frames are placed on a lower layer than forward predicted image layers indicates that for purposes of illustration, backward predicted image frames are in this coding example considered subjectively less valuable than forward predicted image frames. Naturally the sub-sequences could both be placed on the same layer, in which case they would be considered equal, or a backward

predicted sub-sequence could be on the upper layer, in which case it would be considered subjectively more valuable.

[0067] Figures 6b and 6c show some alternatives for coding a video sequence according to Figure 6a. In Figure 6b both forward and backward predicted sub-sequences are placed on the base layer, the I-frame being only located on the INTRA-layer. The forward predicted sub-sequence on this layer is thus the second sub-sequence and its sub-sequence identifier is 1. In Figure 6c, in turn, an I-frame and a forward predicted sub-sequence based on it are located on the base layer, while a backward predicted sub-sequence is located on enhancement layer 1.

[0068] Moreover, according to a preferred embodiment of the invention, the above-described scalability can be utilized for coding what is known as a scene transition into a video sequence. Video material, such as news reports, music videos and movie trailers, often comprise rapid cuts between separate image material scenes. Sometimes the cuts are abrupt, but often a procedure known as scene transition is used in which transfer from one scene to another takes place by dimming, wiping, mosaic dissolving or scrolling the image frames of a previous scene, and, correspondingly, by presenting those of a later scene. From the point of view of coding efficiency, the video coding of a scene transition is often most problematic, because the image frames appearing during the scene transition comprise information on the image frames of both the terminating and the initiating scene.

[0069] A typical scene transition, fading, is carried out by gradually reducing the intensity or luminance of the image frames of a first scene to zero, while gradually increasing the intensity of the image frames of a second scene to its maximum value. This scene transition is referred to as cross-faded scene transition.

[0070] Generally speaking, a computer-made image can be thought of as consisting of layers, or image objects. Each object can be defined with reference to at least three information types: the structure of the image object, its shape and transparency, and the layering order (depth) in relation to the background of the image and to other image objects. Shape and transparency are often determined using what is known as an alpha plane, which measures opacity and the value of which is usually determined separately for each image object, possibly excluding the background, which is usually determined as non-transparent. The alpha plane value of a non-transparent image object, such as

the background, can thus be set at 1.0, whereas the alpha plane value of a fully transparent image object is 0.0. The values in between define the intensity of the visibility of a specific image object in a picture in proportion to the background and to other, at least partly overlapping, image objects that have a higher depth value than the image object in question.

[0071] The superimposition of image objects in layers according to their shape, transparency and depth position is referred to as scene composition. In practice the procedure is based on the use of weighted averages. First, the image object that is closest to the background, i.e. deepest according to its depth position, is placed onto the background and a combined image is formed of the two. The pixel values of the combined image are formed as an average weighted by the alpha plane values of the background image and the image object in question. The alpha plane value of the combined image is then set at 1.0, after which it serves as a background image for the next image object. The process continues until all image objects are attached to the image.

[0072] In the following, a procedure according to a preferred embodiment of the invention will be described in which video sequence scalability layers are combined with the above described image objects of image frames and their information types to provide a scene transition with scalable video coding that also has good compression efficiency.

[0073] This embodiment of the invention is illustrated in the following by way of example and in a simplified manner by using cross-faded scene transition, on one hand, and abrupt scene transition, on the other hand, as examples. The image frames to be displayed during a scene transition are typically formed of two superimposed image frames, a first image frame comprising a first image scene and a second image frame a second scene. One of the image frames serves as the background image and other, which is referred to as a foreground image, is placed on top of the background image. The opacity of the background image, i.e. its non-transparency value, is constant. In other words, its pixel-specific alpha plane values are not adjusted.

[0074] In this embodiment of the invention, the background and foreground images are both defined according to scalability layer. This is illustrated in Figure 7, which shows an example of how image frames of two different scenes can be placed on scalability layers during a scene transition of the invention. Figure 7 shows a first image frame 700 of a first (terminating)

scene positioned on the base layer. The image frame 700 may be either an I-frame containing image information that has not been determined using motion-compensated temporal predicting, or it may be a P-frame that is a motion-compensated image frame predicted from previous image frames. The coding of a second (initiating) scene starts during the temporally following image frame and, according to the invention, the image frames of the scene are also placed on the base layer. Remaining image frames 702, 704 of the second (terminating) scene are then placed on enhancement layer 1. These image frames are typically P-frames.

[0075] In this embodiment, the image frames of the second (initiating) scene are thus placed on the base layer, at least for the duration of the scene transition. The first image frame 706 of the scene is typically an I-frame, and it is used for temporally predicting the succeeding image frames of the scene. Consequently, the succeeding image frames of the second scene are temporally predicted frames, typically P-frames, such as frames 708 and 710 shown in Figure 7.

[0076] According to a preferred embodiment of the invention, this placing of image frames on scalability layers can be used for implementing a cross-faded scene transition by determining the image layer that is on the base layer always as a background image of maximum opacity (100%), or non-transparency value. During a scene transition, image frames located on enhancement layers are placed onto the background image and their opacity is adjusted for example by means of suitable filters such that the frames gradually change from non-transparent to transparent.

[0077] In the video sequence of Figure 7, there are no image frames on the lower scalability layers during the first base layer image frame 700. For this time instant, the first image frame 700 is only coded into the video sequence.

[0078] The next image frame 706 of the base layer initiates a new (second) scene, during which the image frame 706 is provided with depth positioning that places it as the background image, and its opacity value is set to the maximum. Temporally simultaneously with the image frame 706 of the base layer, there is an image frame 702 of a terminating (first) scene on enhancement layer 1. To allow a cross-faded scene transition to be produced, the transparency of the frame 702 must be increased. The example of Figure 7 assumes that the opacity of the image frame 702 is set at 67% and, in addition,

the image frame 702 is provided with depth positioning that determines it as a foreground image. For this time instant, an image combining the image frames 706 and 702 is coded into the video sequence, image 706 being visible as a weaker image on the background and image 702 as a stronger image at the front, because its opacity value is essentially high (67%).

[0079] During the temporally following image frame, there is a second image frame 708 of the second scene on the base layer, the frame 708 being thus correspondingly provided with depth positioning determining it as a background image, and its opacity value is set to the maximum. Enhancement layer 1 further comprises the last image frame 704 of a temporally simultaneously terminating (first) scene, the opacity value of the frame being set at 33% and, in addition, the image frame 704 being provided with depth positioning that determines it as a foreground image as well. Consequently, for this time instant, an image combined of the image frames 708 and 704 is coded into the video sequence, the image 708 being displayed as a stronger image on the background and the image 704 as a weaker image on the foreground, because the opacity value of the image 704 is no longer more than 33%.

[0080] During the temporally following image frame, the base layer comprises a third image frame 710 of the second scene. Since the first scene has terminated, only the image frame 710 is coded into the video sequence, and the displaying of the second scene continues from the frame 710.

[0081] The above disclosure describes, by way of example, the positioning of image frames according to the invention on scalability layers to implement cross-faded scene transition in a manner that is advantageous from the point of view of coding efficiency. However, it is possible that when a video sequence is being transmitted or decoded, a situation arises in which the bit rate of the video sequence must be adjusted according to the maximum value of the bandwidth and/or terminal device decoding rate available for data transfer. This kind of bit rate control causes problems when the scene transition is to be implemented using prior art video coding methods.

[0082] A preferred embodiment of the present invention now allows one or more scalability layers, or independently decodable sub-sequences included in them, to be removed from a video sequence, whereby the bit rate of the video sequence can be decreased and yet, at the same time, the video sequence can be decoded without reducing image frequency. In the image

frame positioning according to Figure 7, this can be implemented by removing enhancement layer 1 from the video sequence. The video sequence is thus only used for displaying the image frames 700, 706, 708 and 710 of the base layer. In other words, a direct transition from the first (terminating) scene to the second (initiating) scene takes place in the form of an abrupt scene transition, i.e. directly from the image frame 700 of the first scene into the I-image frame 706 that initiates the second scene. The transition is thus not a cross-faded scene transition but an abrupt scene transition. Nevertheless, the scene transition can be carried out in an advantageous manner without affecting the quality of the video sequence image, and the viewer usually does not experience an abrupt scene transition carried out instead of a cross-faded scene transition in any way disturbing or faulty. On the contrary, since the prior art implementation does not allow scalability layers to be removed, scene transition would often require image frequency to be reduced, which the viewer would find jerky and disturbing.

[0083] The invention thus provides a preferred means for carrying out multimedia data traffic shaping in a streaming server comprising information about the different sub-sequences of a video sequence: their average bit rate, location in relation to the entire video sequence, duration and their interdependencies regarding the layers. The streaming server also determines the maximum value of the bandwidth available for the data transfer and/or the decoding rate of the terminal device. On the basis of this information, the streaming server decides how many scalability layers and which sub-sequences are transmitted in the video sequence. Bit rate control can thus be carried out, when necessary, by making first a rough adjustment of the number of the scalability layers, after which finer sub-sequence-specific adjustment can be easily carried out. At its simplest, bit rate control means making sub-sequence-specific decisions on whether a particular sub-sequence will be added to a video sequence or removed from it. In case of removal it is advisable to remove entire sub-sequences from a video sequence, because the removal of separate images may cause errors in other images of the same sub-sequence. For the same reason, all sub-sequences of a lower enhancement layer should be left out if they are dependent on the removed sub-sequence of a higher layer. If there are interdependent sub-sequences on one and the same scalability layer, sub-sequences dependent on an earlier sub-sequence must be removed if the earlier sub-sequence is removed.

[0084] If the image frame identifier data are added to a video sequence that is to be transmitted, traffic shaping can also be carried out in a telecommunications network element to be used for the transfer of the video sequence, for example in an Internet router, in different gateways, or at a base station or base station controller of a mobile communications network. For the network element to be able to maintain and process the sub-sequence information, it must have extra memory and processing capacity. For this reason traffic shaping that is to be carried out in the network is perhaps most probably executed using simple processing methods, such as the DiffServ, i.e. differentiated services, procedure that is supported by some IP-based networks. In the DiffServ method, each IP data packet is assigned a priority, whereby data packets of a higher priority are relayed faster and more reliably to the recipient than packets of a lower priority. This is advantageously applied to the scalability of the invention by determining not only scalability-layer-specific, but also sub-sequence-specific priorities, which enables a highly advanced priorisation.

[0085] There are many alternatives for adding image frame identifier data to a video sequence that is to be transmitted. In addition, it is also possible not to include any identifier data into the video sequence, in which case traffic shaping is only carried out at the streaming server. The identifier data can be included in the header fields of a video sequence, or in the header fields of the transfer protocol to be used, such as RTP (Real Time Protocol). According to a preferred embodiment, the identifier data can be transferred using a Supplemental Enhancement Information (SEI) mechanism. SEI provides a data delivery mechanism that is transferred synchronously with the video data content, thus assisting in the decoding and displaying of the video sequence. The SEI mechanism, particularly when used for transferring layer and sub-sequence information, is disclosed more in detail in the ITU-T standard document ITU-T Rec. H.264 (ISO/IEC 14496 – 10:2002), Annex D. In the cases, wherein a separate transfer protocol or mechanism is used for identifier data transfer, traffic shaping can be carried out also at one of the network elements of the transfer path. In addition, the receiving terminal device can control the decoding.

[0086] If the encoder or decoder supports reference picture selection, video sequence coding requires that decoded image frames be buffered before the coding so as to allow the relationships between different

image frames to be temporally predicted from one or more other image frames. Image frame buffering can be arranged at least in two different ways, either as sliding windowing or as adaptive buffer memory control. In sliding windowing, M image frames that were coded last are used as a buffer. The frames in the buffer memory are in a decoded and reconstructed form, which allows them to be used as reference images in the coding. As the coding proceeds, the image frame buffering functions on the basis of the FIFO principle (First-In-First-Out). Images that are not used as reference images, such as conventional B-images, do not need to be stored in the buffer. Alternatively, the buffering can be also be implemented as adaptive buffer memory control, in which case the image buffering is not restricted to the FIFO principle, but image frames that are not needed can be removed from the buffer in the middle of the process, or, correspondingly, some image frames can be stored in the buffer for a longer period of time, if they are needed as reference images for later image frames. A known reference picture selection is implemented by indexing image frames that are in the buffer memory into a specific order, the image indices being then used to refer to an image in connection with motion-compensation, for example. This indexing method generally provides better compression efficiency compared to using image numbers, for example, for referring to a specific image when motion-compensation reference images are to be signalled.

[0087] The above reference image indexing method is sensitive to transfer errors, because the buffers of the sender's encoder and the recipient's decoder must contain mutually corresponding reconstructed images in identical order to ensure that the encoder and decoder both form the same indexing order. If the image frames are indexed in different order in the buffers of the encoder and the decoder, an incorrect reference image may be used in the decoder. To prevent this, it is essential that the decoder can be controlled to take into account image frames and sub-sequences that the encoder has intentionally removed from the video sequence. In that case the image frame numbering may comprise gaps, which the decoder typically interprets as errors and tries to reconstruct the image frames interpreted as lost. For this reason, it is essential that the encoder is capable to inform the decoder that the discontinuities in the image numbering of the transmitted image frames are intentional.

[0088] In response to this, and provided that sliding windowing is used for buffering the image frames, the decoder enters into the buffer memory a number of image frames, the contents of which may be fully random, corresponding to the missing image numbers. These random image frames are then designated by an identifier "invalid" to indicate that the frames in question do not belong to the actual video sequence, but are only filler frames entered for purposes of buffer memory management. A filler frame can naturally be implemented using only memory indicators, i.e. no data is preferably entered into the buffer memory, but memory management is used merely to store a reference to a generic "invalid" frame. The entering of the image frames of the actual video sequence continues from the correct image frame number after the number of filler frames indicated by the missing image numbers has been entered into the buffer, which allows the buffer memories of the encoder and the decoder to be kept preferably in synchronism. If during decoding a reference to an image number is detected which is then found to indicate a filler frame located in the buffer, error correction actions are initiated in the decoder to reconstruct the actual reference image, for example by asking the encoder to re-transmit the reference image in question.

[0089] Further, the procedure of the invention allows separate buffer memories to be used on the different scalability layers, or, correspondingly, sub-sequence-specifically. Each scalability layer may thus have a separate buffer memory that is conceptually separate and functions on the basis of the sliding window principle. Similarly, each sub-sequence may also be provided with a conceptually separate buffer memory that also functions on the basis of the sliding window principle. This means that the buffer memory is always emptied when a sub-sequence terminates. Separate buffer memories can be used in a preferred manner for reducing the need for signalling in certain situations in which ordinary sliding window buffering would be inadequate and actively adaptive buffer memory management would need to be used instead.

[0090] The H.26L standard defines a picture order count as a picture position in output order. The decoding process specified in the H.26L standard uses picture order counts to determine default index orderings for reference pictures in B slices, to represent picture order differences between frames and fields for vector scaling in motion vector prediction and for implicit mode weighted prediction in B slices, and to determine when successive slices

in decoding order belong to different pictures. The picture order count is coded and transmitted for each picture.

[0091] In one embodiment of the invention, the decoder uses the picture order count to conclude that pictures are temporally overlapping, i.e., pictures that have an equal picture order count are temporally overlapping. Preferably, the decoder outputs only the picture on the highest received layer. In the absence of layer information, the decoder concludes that the latest temporally overlapping picture in decoding order resides on highest received layer.

[0092] The above disclosure describes a procedure for coding video frames for the purpose of producing a scalable, compressed video sequence. The actual procedure is carried out in a video encoder, such as the compressor 108 of Figure 1, which may be any known video encoder. For example a video encoder according to the ITU-T recommendation H.263 or H.26L may be used, the video encoder being arranged to form, in accordance with the invention, a first sub-sequence into a video sequence, at least part of the sub-sequence being formed by coding I-frames; to form at least a second sub-sequence into the video sequence, at least part of the sub-sequence being formed by coding at least P- or B-frames, and at least one video frame of the second sub-sequence being predicted from at least one video frame of the first sub-sequence; and to determine into the video sequence the identification data of at least the video frames of the second sub-sequence.

[0093] According to the procedure of the invention, each sub-sequence of a particular scalability layer is preferably independently decodable, naturally taking into account dependencies from higher scalability layers and possibly other sub-sequences of the same scalability layer. A scalably compressed video sequence such as the one described above can thus be decoded by decoding a first sub-sequence of a video sequence, at least part of the sub-sequence having been formed by coding at least I-frames, and by decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence having been formed by coding at least P- or B-frames, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence, and by determining the identification and dependency data of at least the video frames comprised by the second sub-sequence of the video sequence, and by

reconstructing at least part of the video sequence on the basis of the sub-sequence dependencies.

[0094] The actual decoding takes places in the video decoder, such as the decompressor 118 of Figure 1, which may be any known video decoder. For example, a low bit rate video decoder according to the ITU-T recommendation H.263 or H.26L may be used, which in this invention is arranged to decode a first sub-sequence of a video sequence, at least part of the sub-sequence having been formed by coding I-frames; to decode at least a second sub-sequence of the video sequence, at least part of the second sub-sequence having been formed by coding at least P- or B-frames and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence. The video decoder is arranged to determine the identification and dependency data of at least the video frames comprised by the second sub-sequence of the video sequence and to reconstruct at least part of the video sequence on the basis of the dependencies of the sub-sequences.

[0095] An essential aspect in the operation of the streaming system of the invention is that the encoder and decoder are positioned at least so that the encoder is operationally connected to the streaming server and the decoder is operationally connected to the receiving terminal device. However, the different elements of the streaming system, terminal devices in particular, may include functionalities that allow two-way transfer of multimedia files, i.e. transmission and reception. The encoder and decoder can thus be implemented in the form of what it known as a video codec integrating both encoder and decoder functionalities.

[0096] It is to be noted that according to the invention the functional elements of the above described streaming system and its elements, such as the streaming server, video encoder, video decoder and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two. The coding and decoding methods of the invention are particularly suitable for implementation as computer software comprising computer-readable commands for executing the process steps of the invention. A preferred way of implementing the encoder and the decoder is to store them in a storage means as a program code that can be executed by a computer-like device, for example a personal computer (PC) or a mobile station, to provide coding/decoding functionalities on the device in question.

[0097] Another alternative is to implement the invention as a video signal comprising a scalably compressed video sequence which in turn comprises video frames coded according to at least a first and a second frame format, the video frames according to the first frame format being independent of other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames. According to the invention, the video signal in question comprises at least a first sub-sequence, at least part of which has been formed by coding at least video frames of the first frame format; at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format; and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and at least one data field that determines video frames belonging to the second sub-sequence.

[0098] It is apparent to a person skilled in the art that as technology advances the basic idea of the invention can be implemented in various ways. The invention and its embodiments are therefore not restricted to the above examples, but they may vary within the scope of the claims.

CLAIMS

1. A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising the steps of

forming into the video sequence at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format; and

forming into the video sequence at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence; and

forming into the video sequence information indicating which video frames belong to the second sub-sequence.

2. A method according to claim 1, **characterized** by comprising the steps of

determining into the video sequence the interdependency between at least the video frames of the second sub-sequence and at least one video frame of the first sub-sequence.

3. A method according to claim 2, **characterized** by comprising the steps of

coding the video sequence into a plural number of scalability layers; and

determining the dependency of the video frames of the second sub-sequence such that at least one video frame of the second sub-sequence is predicted from a group comprising

a video frame of a higher scalability layer,

a video frame of another sub-sequence in the same scalability layer.

4. A method according to claim 3, **characterized** by comprising the step of

determining the dependencies of the video frames of the second sub-sequence on the basis of at least a scalability layer identifier and a sub-sequence identifier.

5. A method according to claim 3 or 4, **characterized** by comprising the step of

coding the first scalability layer of the video sequence to comprise video frames according to one frame format, each one of the frames forming a separate sub-sequence.

6. A method according to claim 3 or 4, **characterized** by comprising the step of

coding the first scalability layer of the video sequence to comprise video frames according to both the first and the second frame format.

7. A method according to claim 3, **characterized** by comprising the step of

determining a unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number.

8. A method according to claim 3, **characterized** by comprising the step of

determining a unique identifier for each video frame on the basis of the image number.

9. A method according to claim 7 or 8, **characterized** by comprising the step of

adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence.

10. A method according to any one of the preceding claims, **characterized** in that

the video frames of the first frame format are I-frames and the video frames of the second frame format are temporally forward and/or backward predicted P-frames, which have been predicted using at least one reference image.

11. A method according to any one of the preceding claims, **characterized** by comprising the step of

coding the sub-sequences in such a way that at least some of the sub-sequence frames are temporally at least partly overlapping.

12. A method according to any one of the preceding claims, **characterized** by comprising the step of

coding the video frames in such a way that the temporal predicting taking place between the video frames is block- or macroblock-specific.

13. An element in a streaming system, such as a streaming server or a network element in a telecommunications system, which is arranged to transmit compressed video sequences to at least one terminal device and to control the bit rate of the video sequence transmission, **characterized** in that

the element is arranged to control the bit rate of a video sequence transmission in which a first sub-sequence of the video sequence comprises video frames of at least a first frame format and at least a second sub-sequence comprises video frames of at least a second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

the bit rate control being arranged to be carried out in at least one of the following ways:

at least one sub-sequence is removed from the video sequence,

at least one sub-sequence is added to the video sequence.

14. A video encoder for forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video encoder is arranged to

form into the video sequence a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format; and to

form into the video sequence at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and to

form into the video sequence information indicating which video frames belong the second sub-sequence.

15. A method for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the

other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** by comprising the steps of

decoding a first sub-sequence of the video sequence, at least part of the first sub-sequence being formed by coding at least video frames of the first frame format;

decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence being formed by coding at least video frames of the second frame format, at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determining dependency data relating at least to the video frames comprised by the second sub-sequence of the video sequence; and

reconstructing at least part of the video sequence on the basis of the dependencies of the sub-sequences.

16. A method according to claim 15, wherein video frames are entered into a sliding buffer memory in connection with decoding, **characterized** by comprising the steps of

decoding from the video sequence an indication informing that the discontinuities in the numbering of the images of the image frames in the video sequence are intentional;

configuring, in response to the indication, the buffer memory to comprise a number of image frames corresponding to the missing image numbers; and

continuing the entering of the image frames comprised by the video sequence in question into the buffer memory from the correct image frame number after the buffer memory has been configured to comprise the number of image frames corresponding to the missing image numbers.

17. A method according to claim 16, **characterized** by comprising the step of

entering into the buffer memory a number of filler frames corresponding to the missing image numbers.

18. A method according to any one of claims 15 to 17, **characterized** by comprising the step of

decoding the video sequence by removing at least one independently decodable sub-sequence from the video sequence.

19. A method according to any one claims 15 to 18, **characterized** by comprising the steps of

initiating the decoding from a random point in the video sequence;

determining an independently decodable sub-sequence that is next after the random point, and

setting the value of the image number of the first video frame in the sub-sequence at zero.

20. A method according to any one claims 15 to 19, **characterized** by comprising the steps of

identifying at least partly temporally overlapping sub-sequence frames from the video sequence based on the picture order count information, and

outputting from the decoder the last image frame in the decoding order, said frame being selected from a group of said at least partly temporally overlapping image frames.

21. A video decoder for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the video decoder is arranged to

decode a first sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the first frame format;

decode at least a second sub-sequence of the video sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence;

determine the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

reconstruct at least part of the video sequence on the basis of the dependency of the sub-sequences.

22. A computer program for coding video frames so as to form a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the

other video frames, **characterized** in that the computer program comprises

a program code for forming a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for forming at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining data into the video sequence to determine which video frames belong to the second sub-sequence.

23. A computer program for decoding a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames, **characterized** in that the computer program comprises

a program code for decoding a first sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the first frame format; and

a program code for decoding at least a second sub-sequence of the video sequence, at least part of the sub-sequence being formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

a program code for determining the dependency data of at least the video frames comprised by the second sub-sequence of the video sequence; and

a program code for reconstructing at least part of the video sequence on the basis of the dependency of the sub-sequences.

24. A video signal comprising a scalably compressed video sequence comprising video frames coded according to at least a first and a second frame format, the video frames of the first frame format being independent of the other video frames, and the video frames of the second

frame format being predicted from at least one of the other video frames, **characterized** in that the video signal comprises

a first sub-sequence, at least part of which is formed by coding at least video frames of the first frame format;

at least a second sub-sequence, at least part of which is formed by coding at least video frames of the second frame format, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and

at least one data field determining the video frames that belong to the second sub-sequence.

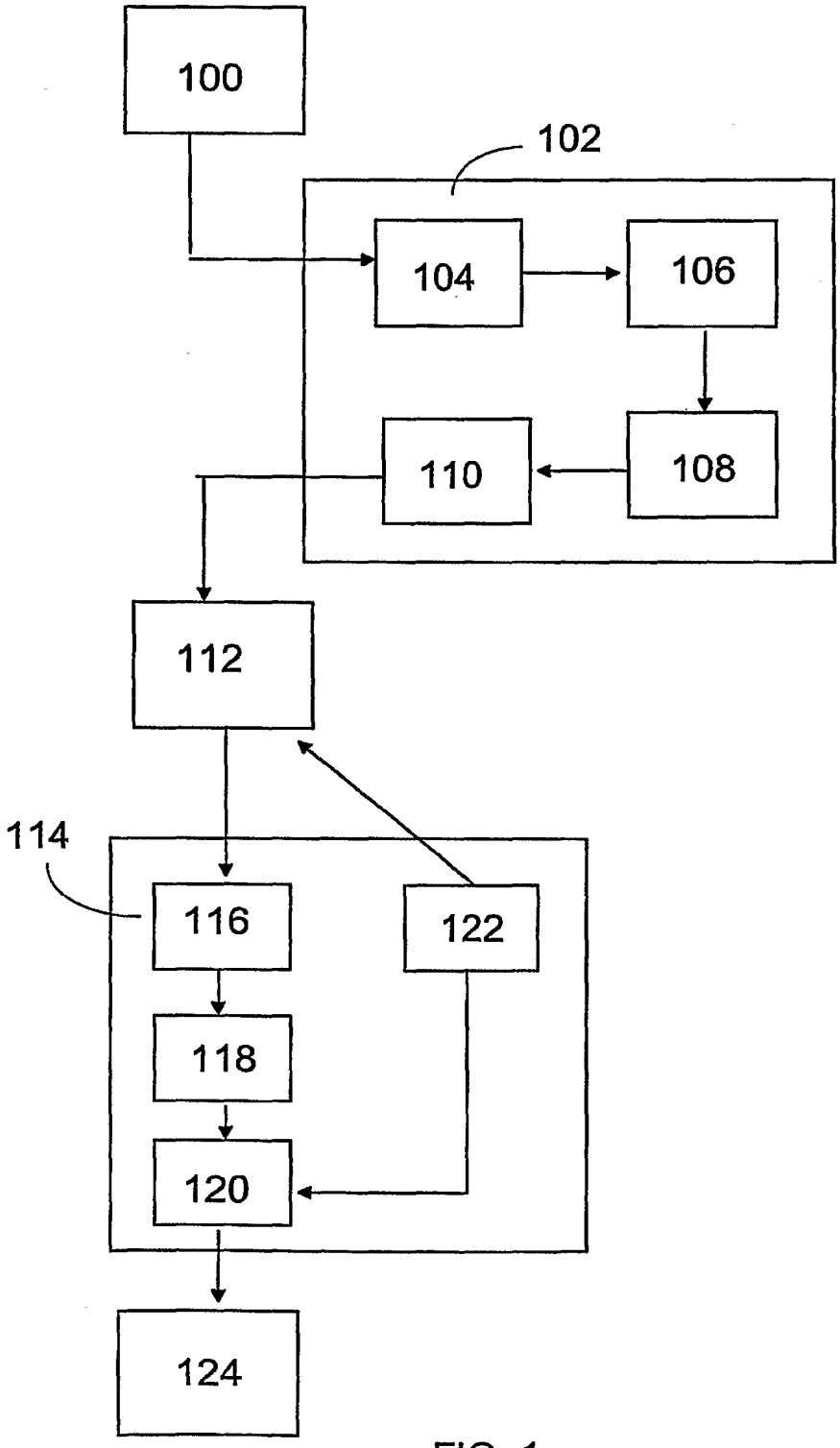


FIG. 1

2/4

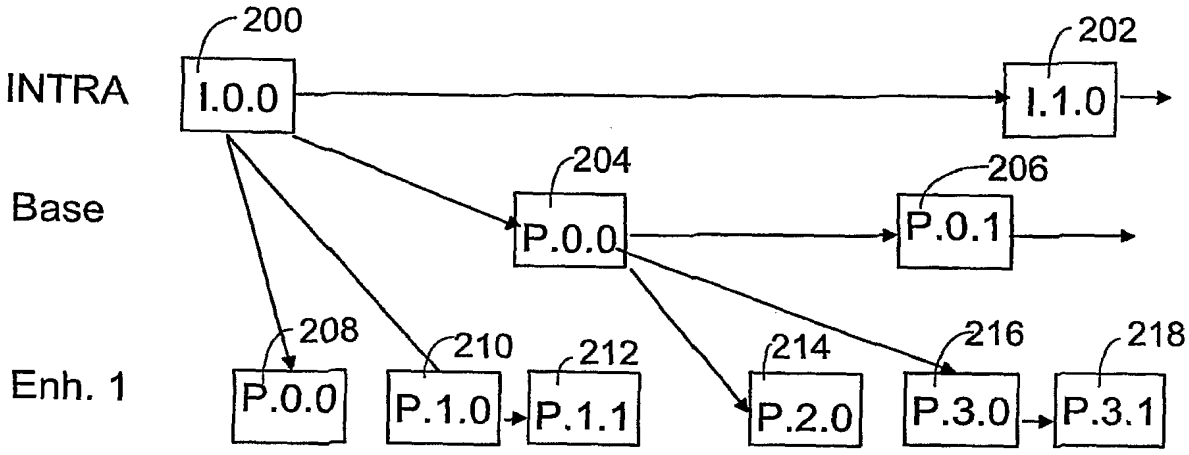


FIG. 2

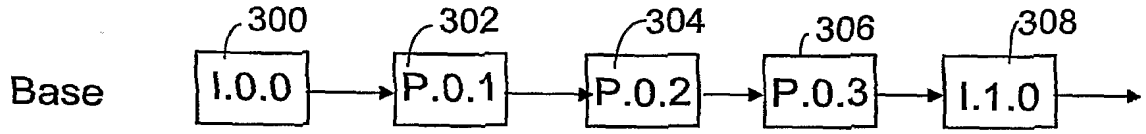


FIG. 3a

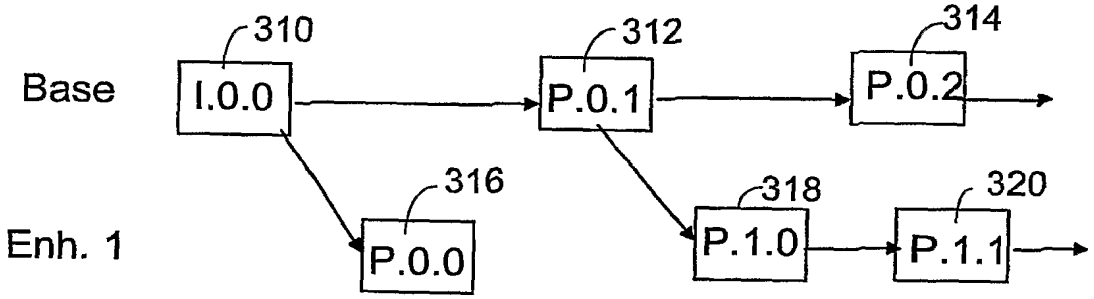


FIG. 3b

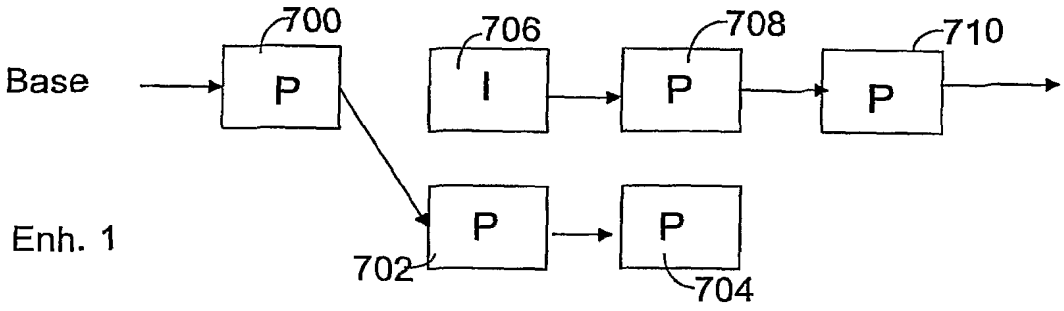


FIG. 7

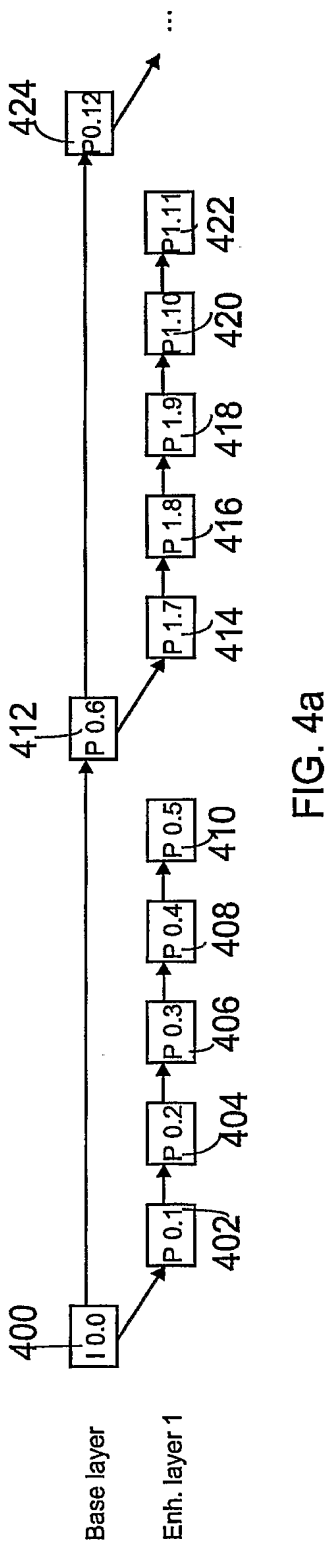


FIG. 4a

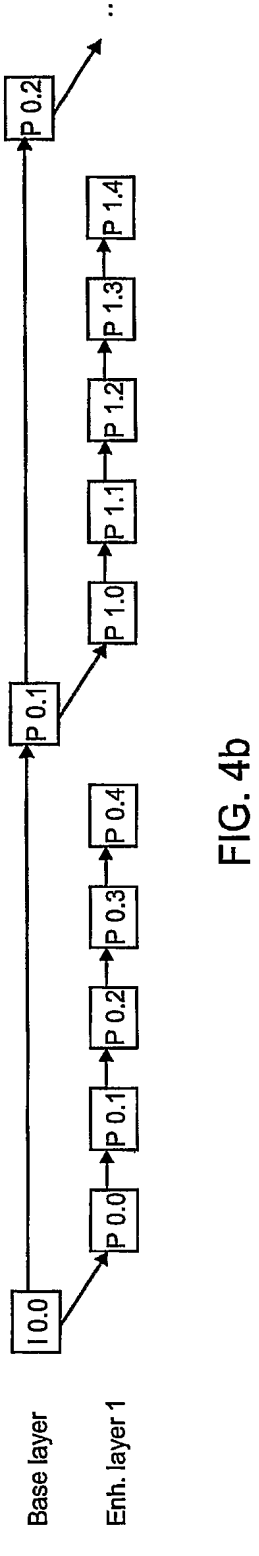


FIG. 4b

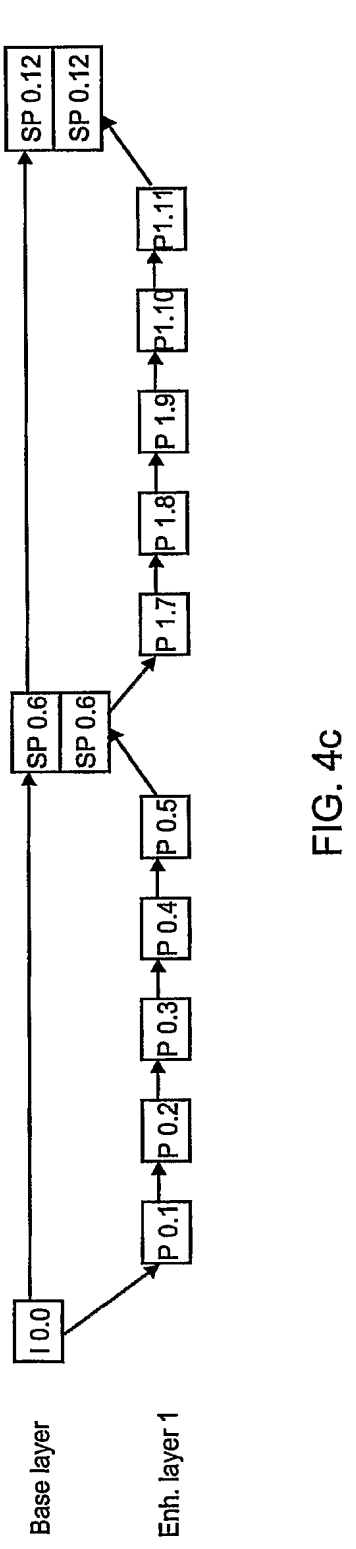


FIG. 4c

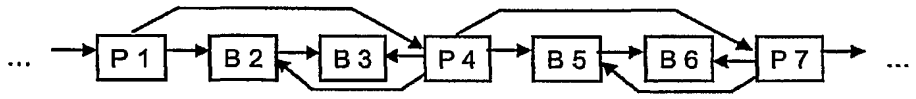


FIG. 5a

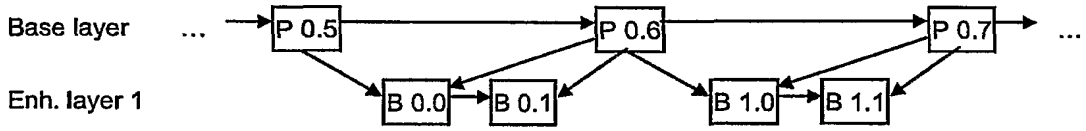


FIG. 5b

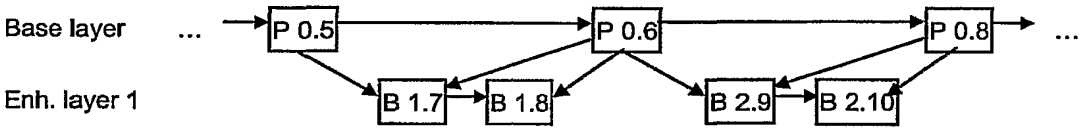


FIG. 5c

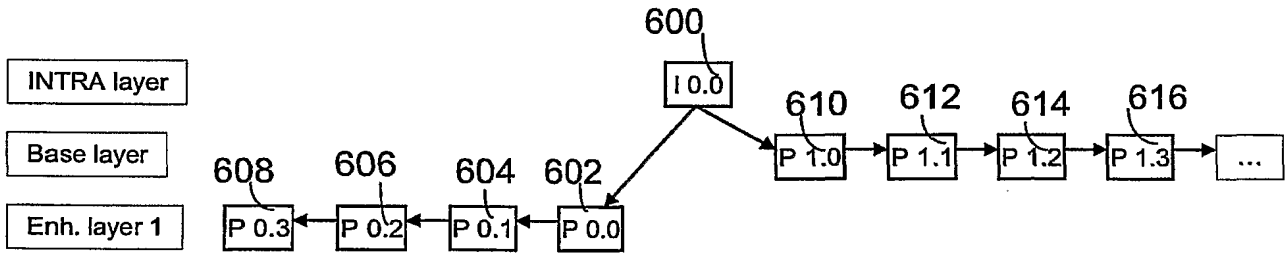


FIG. 6a

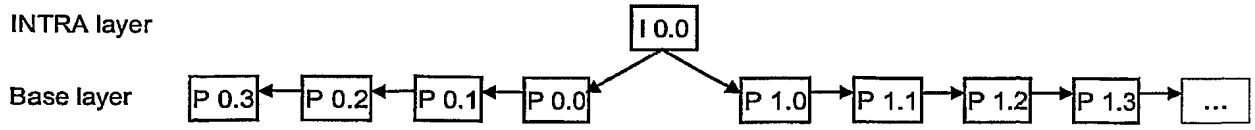


FIG. 6b

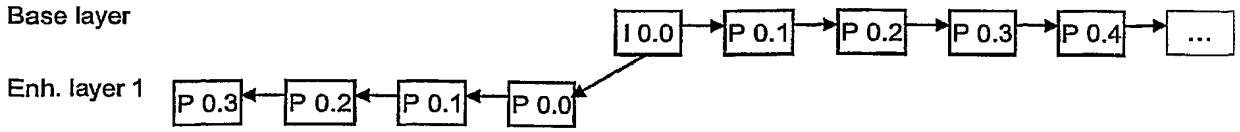


FIG. 6c

Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG
(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)
4th Meeting: Klagenfurt, Austria, 22-26 July, 2002

Document: JVT-D098
Filename: JVT-D098.doc

Title:	**Signaling of Enhanced GOPs**
Status:	Input Document to JVT
Purpose:	Proposal

Author(s) or Contact(s):
Miska M. Hannuksela
Nokia Mobile Software
P.O. Box 88
33721 Tampere
Finland

Tel: +358 40 521 2845
Email: miska.hannuksela@nokia.com

Source: Nokia Corporation

---

1. INTRODUCTION

The enhanced GOP concept introduced in JVT-B042 was adopted to the JVT coding standard in the second JVT meeting. The enhanced GOP concept organizes pictures in layers and sub-sequences. A sub-sequence represents a number of inter-dependent pictures that can be disposed without affecting the decoding of any other sub-sequence in less enhanced layers. The enhanced GOP concept was proposed for unequal error protection and rate shaping purposes in streaming servers. Consequently, the concept required support only in the file storage format. JVT-B042 included the changes regarding the interim file format in the working draft. No changes in the JVT syntax were seen necessary at that time.

JVT-C080 proposed a couple of applications that could utilize transmission of enhanced GOP information. However, the JVT did not consider these applications as important enough to justify the inclusion of the enhanced GOP feature in the syntax and semantics of the NAL units. Since the Fairfax meeting, we found a couple of other use cases where the transmitted enhanced GOP information would be beneficial. We present these use cases in this contribution.

The contribution is organized as follows: Section 2 summarizes the use cases for signaling of layers and sub-sequences within the coded data stream. Section 3 presents the proposed changes relative to the JVT CD. Section 4 presents one of the uses, namely the Multimedia Messaging Service.

Proposed agenda category: improvements & enhancements

Keywords for agenda allocation: high-level syntax and semantics

2. USE CASES FOR ENHANCED GOP SIGNALING

We have identified the following applications for signaling of layers and sub-sequences within the coded data stream:

- Identification of coded data in out-of-decoding-order transmission and recovery of decoding order from transmission order. The out-of-decoding-order transmission can be used to smooth out bit-rate and delay variations, and to have more time for

retransmissions of subjectively most important data in streaming systems. See JVT-D093 for details.

- Identification of random access, splicing, and editing operations. See JVT-D097 for details.
- Signaling the reference frames of sync pictures as explained in JVT-D101.
- Signaling of bit-rate-scalable bit-streams when transmitting a file from an encoder to a streaming server. This can be utilized in the Multimedia Messaging Service (MMS) of mobile networks, for example. See section 4 for details.
- Computational scalability in decoders. As explained in JVT-C080, sometimes there is no file or stream suitable for decoder's computational capabilities available, while a very demanding stream or file has to be used. In multitasking systems the processing power is shared between multiple processes, and therefore the performance of the video decoder may fluctuate depending on the number and complexity of other tasks executed simultaneously.
- A video communication gateway may sometimes have to tune the bit-rate of a coded video stream. The abilities of bit-rate control methods for coded data are limited. Thus, it is advantageous to carry enhanced GOP information in systems where bit-rate control of coded video is desired. Gateways may decide to forward only certain layers or they may decide to discard one sub-sequence to drop the bit-rate.

3. PROPOSED CHANGES IN THE JVT CD

3.1 Definitions

The following definitions are to be inserted to section 3 of the JVT standard:

3.QQ sub-sequence layer: A sub-sequence layer contains a subset of the pictures in a coded data stream. A sub-sequence layer is characterized by an average bit-rate and an average frame rate. Sub-sequence layers can be ordered hierarchically based on their dependency on each other, i.e., layer 0 is independently decodable, pictures in layer 1 may be predicted from layer 0, pictures in layer 2 may be predicted from layers 0 and 1, etc. The subjective quality increases along with the number of decoded layers. A layer having a larger layer number is a higher layer than a layer having a smaller layer number.

3.QQ sub-sequence: A set of pictures that cover a certain period of time within the sequence. The presentation times of pictures in different sub-sequences can overlap partly or entirely. A picture shall reside in one layer and in one sub-sequence only. A sub-sequence shall not depend on any other sub-sequence in the same or in a higher sub-sequence layer. A sub-sequence in layer 0 can be decoded independently of any other sub-sequences and previous long-term reference pictures.

3.2 Parameter Set

The following parameter is added to the parameter set:

num_layers	0	e(v)

num_layers 0 indicates that no layer number or sub-sequence identifier is present in slice headers. Value 1 indicates that no layer number is present in slice headers but sub-sequence identifiers are included in slice headers. Values greater than 2 indicate that the layer number field is present in slice headers.

3.3 Picture-Layer Data

The following parameters are added to the picture layer. If there will be no picture layer in the JVT standard, the parameters may either reside in the slice header or in the parameter set.

if (num_layers > 1)		
layer_num	3	e(v)
if (num_layers > 0)		
subsequence_id	3	e(v)

layer_num indicates the sub-sequence layer number of the picture.

subsequence_id identifies the sub-sequence within a layer. Consecutive sub-sequences in transmission order shall have a different subsequence_id from each other.

3.4 Supplemental Enhancement Information

The characteristics of layers and sub-sequences as well as dependency information of sub-sequences may be used in sub-sequence-based editing operations. We think that Supplemental Enhancement Information is a correct transport mechanism for this type of information. The presented notation is not aligned to the notation of the JVT CD due to lack of time, but a straightforward conversion can be easily done later.

Layer Information

```
while (more_data_in_SEI_payload()) {
   unsigned int(32) avgBitRate;
   unsigned int(32) avgFrameRate;
}
```

The structure above is repeated until the size of the SEI payload is reached. The first instance of the structure signals the data for layer 0. The second instance of the structure signals the data for layer 1 and so on.

avgBitRate gives the average bit-rate in bits/second of the layer. All NAL units of the layer are taken into account in the calculation. Value zero means an undefined bit-rate.

avgFrameRate gives the average frame rate in frames/(256 seconds) of the layer. Value zero means an undefined frame rate.

The values are in effect from the point they are transmitted until an update of the values is transmitted.

Sub-Sequence Information

```
{
   uvlc layerNumber;
   uvlc subSequenceIdentifer;
   uvlc pictureNumber;
```

```
   uvlc parameterSetAddress;
   unsigned int(64) ssDuration;
   unsigned int(32) avgBitRate;
   unsigned int(32) avgFrameRate;
   uvlc numReferencedSubSequences;
   for (i = 0; i < numReferencedSubSequences; i++) {
      uvlc layerNumber;
      uvlc subSequenceIdentifier;
}
```

One structure as above exists per one SEI message.

layerNumber and subSequenceIdentifier identify the sub-sequence.

pictureNumber identifies the first picture of the sub-sequence.

ssDuration indicates the duration of the sub-sequence. ssDuration equal to zero indicates an undefined duration. The clock frequency for ssDuration is fetched from the parameter set indicated by parameterSetAddress.

avgBitRate gives the average bit-rate in bits/second of the sub-sequence. All NALPs of the sub-sequence are taken into account in the calculation. Value zero means an undefined bit-rate.

avgFrameRate gives the average frame rate in frames/(256 seconds) of the sub-sequence within the segment. Value zero means an undefined frame rate.

numReferencedSubSequences gives the number of directly referenced sub-sequences. The following loop lists the referenced sub-sequences, i.e., layerNumber and subSequenceIdentifier within the loop identify a sub-sequence that is used as a motion compensation reference for the current sub-sequence.

3.5 Decoder Process

Normative decoding process for some of the applications has been proposed in JVT-D093, JVT-D097, and JVT-D101. For the other presented applications, the decoder process is non-normative. The following text may be inserted to a non-normative part of the JVT standard:

The decoder may decide not to decode a certain layer having layer number n. In such a case, the decoder shall not decode any slice having a layer number greater than n and any slice in other sub-sequence that has a same layer number.

The decoder may decide not to decode a certain sub-sequence lies in layer n and have sub-sequence identifier m. In practice, the decoder shall not decode any slice having a layer number equal to n and a sub-sequence identifier equal to m. Once the decoder has received a Sub-Sequence Information SEI message concerning a sub-sequence in layer n and having the sub-sequence identifier m but starting from a different picture number than the discarded sub-sequence, the decoder should stop discarding slices having layer number n and sub-sequence identifier m.

3.6 Other Syntax Design Alternatives

It could be argued that layer numbers and sub-sequence identifiers could be conveyed in SEI messages. However, because some of the identified use cases require normative decoder process related to sub-sequence identifiers and because standard-compliant decoders may not support SEI messages, the SEI mechanism is not suitable.

Sub-sequences can but do not have to be organized into layers. Layers help to signal long-term characteristics of sub-sequence series. They also help signaling sub-sequence dependencies, as a sub-sequence can only depend on sub-sequences in lower layers. However, there is nothing that would mandate the use of layer numbers in slice headers, and therefore the layer number could be signaled as a part of an appropriate SEI message.

4. MULTIMEDIA MESSAGING SERVICE

This section presents the Multimedia Messaging Service (MMS), discusses video coding requirements for the service, and justifies why the enhanced GOP concept suits MMS.

4.1 Overview

The Multimedia Messaging Service (MMS) can be considered a continuation of the famous Short Message Service (SMS) of GSM mobile networks.

A simplified flow of operations in an MMS transaction is as follows: A user creates a message. For example, he can shoot a video clip with his mobile camera phone. After creating the message, he selects a receiver and sends the message. In practice, an audio-video message is stored according to the 3GPP flavor of the ISO media file format (.3GP) and uploaded to an MMS server using a reliable transport protocol. Then, depending on the capabilities of the receiver terminal, user preferences, and the capabilities of MMS network elements, the message is either downloaded or streamed to the receiving terminal.

MMS works in both 2G (GSM and its evolutions) and 3G mobile networks. Consequently, MMS streaming may take place through different types of mobile networks, such as GPRS or UMTS, which are further characterized by a variety of characteristics in the radio access network. Notice also that the downlink network characteristics are unknown when the message is being created. Moreover, the characteristics of the mobile network may change in a function of the location of the terminal and the number of terminals in a cell among other things. Consequently, the network throughput may be highly varying. In order to stream in real-time through a variable bit-rate connection, the MMS server should be able to adjust the media bit-rate easily.

It is expected that a majority of multimedia messages will be created in hand-held terminals. Generally, at least the following aspects of hand-held terminals limit the possibilities when creating multimedia messages:

- Processing power is limited compared to desktop computers.
- The amount of data and program memory is limited compared to desktop computers.
- Screen size is small compared to desktop computers.
- User interaction is typically limited to a relatively small keyboard.

4.2 JVT Video Coding for MMS

We believe that the JVT codec has good chances to become a mandatory video codec in a future evolution of MMS. Thus, it would be beneficial if the JVT codec provide tools that make it more suitable for MMS (compared to previous video coding standards, for example).

Due to the presented characteristics of MMS, we consider the following as key features of a video encoder:

- One-pass causal or nearly causal encoding in order to avoid storage of uncompressed frames in a device having limited storage space. (Herein, “causal” refers to encoding of a frame immediately after it is captured.)
- The encoder should output bit-rate-scalable files allowing the MMS server to shape the stream for different network conditions. The dynamic range of scalability should be large. We expect that a practical range of bit-rates is from 10 to 64 kbps in the short run and the range is likely to become broader in the long run.
- Fast encoding.
- Efficient compression.

One of the few bit-rate-scalability tools included in the JVT syntax is the possibility to discard non-stored frames. Conventional B frames provide efficient means for bit-rate-scalable coding, but their relevance decreases in low frame rates (used in very low bit rates expected in MMS), i.e., their compression efficiency drops in low frame rates. Conventional B frames require storage of uncompressed frames, their encoding requires more processing power and more memory than encoding of I and P frames, and the dynamic range of achievable bit-rates is limited, typically up to 50 % bit-rate reduction from the total bit-rate is achievable (in 25-30 Hz frame rates). Thus, we feel that other means for bit-rate-scalability would be desirable.

Encoders can achieve larger dynamics in bit-rate-scalability and use less memory than with conventional B frames when coding streams similar to the one depicted in Figure 1. “SP 6” (and “SP 12”) boxes indicate identical representations of frame 6 (and 12 respectively) based on different reference frames. A streaming server decides which coded SP frame is transmitted based on the temporal resolution of the previously transmitted section of the bitstream. The encoder may switch to a lower or higher temporal resolution after such an SP frame. The compression efficiency of these kinds of coding schemes is nearly as good as in a non-scalable coding scheme as shown in VCEG-M73.

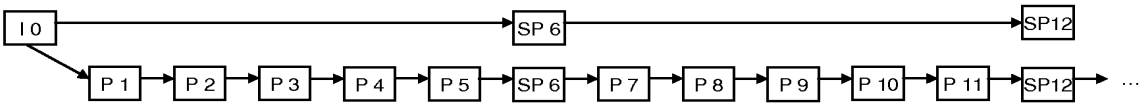


Figure 1. Temporal scalability for rate shaping.

It was shown in JVT-B042 that the enhanced GOP concept is optimal for signaling the dependencies of coding scenarios such as the one shown in Figure 1. As audio-video files are uploaded using the ISO media file format, the enhanced GOP concept should be ideally conveyed as metadata in the file format. Thus, in order to ensure the support of enhanced GOP concept in MMS, we propose adding the enhanced GOP signaling possibility into the JVT syntax.

JVT Patent Disclosure Form

International Telecommunication Union
Telecommunication Standardization Sector

International Organization for Standardization

International Electrotechnical Commission

Joint Video Coding Experts Group - *Patent Disclosure Form*

(Typically one per contribution and one per Standard | Recommendation)

Please send to:
JVT Rapporteur Gary Sullivan, Microsoft Corp., One Microsoft Way, Bldg. 9, Redmond WA 98052-6399, USA
Email (preferred): Gary.Sullivan@itu.int Fax: +1 425 706 7329 (+1 425 70MSFAX)

This form provides the ITU-T | ISO/IEC Joint Video Coding Experts Group (JVT) with information about the patent status of techniques used in or proposed for incorporation in a Recommendation | Standard. JVT requires that all technical contributions be accompanied with this form. *Anyone* with knowledge of any patent affecting the use of JVT work, of their own or of any other entity ("third parties"), is strongly encouraged to submit this form as well.

This information will be maintained in a "living list" by JVT during the progress of their work, on a best effort basis. If a given technical proposal is not incorporated in a Recommendation | Standard, the relevant patent information will be removed from the "living list". The intent is that the JVT experts should know in advance of any patent issues with particular proposals or techniques, so that these may be addressed well before final approval.

This is not a binding legal document; it is provided to JVT for information only, on a best effort, good faith basis. Please submit corrected or updated forms if your knowledge or situation changes.

This form is *not* a substitute for the *ITU ISO IEC Patent Statement and Licensing Declaration*, which should be submitted by Patent Holders to the ITU TSB Director and ISO Secretary General before final approval.

Submitting Organization or Person:	
Organization name	**Nokia Corporation**
Mailing address	
Country	
Contact person	
Telephone	
Fax	
Email	
Place and date of submission	
**Relevant Recommendation	Standard and, if applicable, Contribution:**	
Name (ex: "JVT")	**JVT**
Title	**Signaling of Enhanced GOPs**
Contribution number	**JVT-D098**

(Form continues on next page)

Disclosure information – Submitting Organization/Person (choose one box)

☐ 2.0 The submitter is not aware of having any granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution.

or,

The submitter (Patent Holder) has granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution. In which case,

☐ 2.1 The Patent Holder is prepared to grant – on the basis of reciprocity for the above Recommendation | Standard – a free license to an unrestricted number of applicants on a worldwide, non-discriminatory basis to manufacture, use and/or sell implementations of the above Recommendation | Standard.

☐ 2.2 The Patent Holder is prepared to grant – on the basis of reciprocity for the above Recommendation | Standard – a license to an unrestricted number of applicants on a worldwide, non-discriminatory basis and on reasonable terms and conditions to manufacture, use and/ or sell implementations of the above Recommendation | Standard.

Such negotiations are left to the parties concerned and are performed outside the ITU | ISO/IEC.

☒ 2.2.1 The same as box 2.2 above, but in addition the Patent Holder is prepared to grant a "royalty-free" license to anyone on condition that all other patent holders do the same.

☐ 2.3 The Patent Holder is unwilling to grant licenses according to the provisions of either 2.1, 2.2, or 2.2.1 above. In this case, the following information must be provided as part of this declaration:

- patent registration/application number;
- an indication of which portions of the Recommendation | Standard are affected.
- a description of the patent claims covering the Recommendation | Standard;

*In the case of any box **other than 2.0** above, please provide the following:*

Patent number(s)/status _____

Inventor(s)/Assignee(s) _____

Relevance to JVT _____

Any other remarks: _____

(please provide attachments if more space is needed)

(form continues on next page)

Third party patent information – fill in based on your best knowledge of relevant patents granted, pending, or planned by other people or by organizations other than your own.

Disclosure information – Third Party Patents (choose one box)

☒ 3.1 The submitter is not aware of any granted, pending, or planned patents *held by third parties* associated with the technical content of the Recommendation | Standard or Contribution.

☐ 3.2 The submitter believes third parties may have granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution.

For box 3.2, please provide as much information as is known (provide attachments if more space needed) - JVT will attempt to contact third parties to obtain more information:

3rd party name(s) _____

Mailing address _____
Country _____
Contact person _____
Telephone _____
Fax _____
Email _____
Patent number/status _____
Inventor/Assignee _____
Relevance to JVT _____

Any other comments or remarks:

Electronic Acknowledgement Receipt

EFS ID:	9810136
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	10888
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	05-APR-2011
Filing Date:	25-JAN-2006
Time Stamp:	18:54:27
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-13-TPO004C1.pdf	58616 d45971109b6c3298745418ede8628e6855d10731	no	2

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	1449-13-TPO004C1.pdf	59877 92566439be7bf06e70c800b875c9ac9c6110af79	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	FH-EP037003217.pdf	11489599 eb3fbc40a44f1218f89d4b32e1668595b2498987	no	222
Warnings:					
Information:					
4	NPL Documents	Hannuksela-2.pdf	73745 d1af007096e859d2cd6d5cc1d112dbf8df536262	no	9
Warnings:					
Information:					
		Total Files Size (in bytes):	11681837		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. § 1.97(c)(1)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted after the mailing date of the first Office action on the merits but before the mailing date of a final Office action under 37 C.F.R. § 1.113 or a Notice of Allowance under 37 C.F.R. § 1.311.

The items listed on the enclosed Form 1449 were first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449. However, if fees are deemed necessary, authorization is given to charge Deposit Account 50-3581 (TPO.004.C1) for consideration of the information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), a copy of the non-U.S. patent/application documents listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that the reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. § 1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,
HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti
Reg. No. 57,125

Application Number	Application/Control No.	Applicant(s)/Patent under Reexamination	
	11/338,996	HANNUKSELA, MISKA	

Document Code - DISQ	Internal Document – DO NOT MAIL

TERMINAL DISCLAIMER	☐ APPROVED	☒ DISAPPROVED
Date Filed : 1/31/11	This patent is subject to a Terminal Disclaimer	

Approved/Disapproved by:

Felicia D. Roberts

See TD Checklist for explanation

U.S. Patent and Trademark Office

Rev. 05/19/09	Doc. Code: DISQ.CKLIST

TERMINAL DISCLAIMER INFORMAL CHECKLIST

APPL. S.N.: 11/338,996	**DATE:**
EXAMINER:	**ART UNIT:**
PARALEGAL: /FELICIA ROBERTS/	**MAIL ROOM DATE:**
NUMBER OF TD(s) FILED: 1	

INSTRUCTIONS: The paralegal has reviewed the submitted TD with the results as set forth below.
If you agree, please use the appropriate form paragraphs identified by this informal memo in your next Office action to notify applicant about the TD. If you disagree, please contact a QAS.

THIS CHECKLIST IS AN INFORMAL, INTERNAL CHECKLIST ONLY. IT MUST NOT BE MAILED TO APPLICANT. IT WILL BE SOFT SCANNED AND NOT VIEWABLE TO THE PUBLIC.

☐ The TD is PROPER and has been accepted and recorded. (See FP 14.23.)

☒ The TD is NOT PROPER and has not been accepted for the reason(s) checked below. (See FP 14.24.)

☐ The disclaimer fee under 37 CFR 1.20(d) in the amount of $ has not been submitted, nor is there any pre authorization in the application to charge to a deposit account. (See FP 14.24 and 14.26.07.)

☐ The LIE has not processed fee for TD (the Paralegal should ask LIE to process the fee).

☐ The TD does not satisfy 37 CFR 1.32(b) (3) in that the person who signed the TD has not stated either: (a) the extent of his/her ownership interest, or (b) the extent of the business/organization entity's ownership interest on whose behalf the person signed. (See FPs 14.26 and 14.26.01.)

☐ The TD lacks the – enforceable only during the period of common ownership – clause needed to overcome a double patenting 37 CFR 1.321(c). (See FP 14.27.01).

☐ The TD lacks 37 CFR 1.321(d) statement for joint research agreement under 35 U.S.C. 103(c) (2) & (3). It doesn't include the waiver and enforceability provisions of 37 CFR 1.321(d). (See FP 14.27.011.)

☐ TD is directed to a particular claim(s); this is not acceptable, since the disclaimer must be of a terminal portion of the entire patent to be granted, MPEP 1490. (See FPs 14.26 and 14.26.02).

☒ The person who signed the terminal disclaimer:

- ☐ failed to state his/her capacity to sign for the business/organization entity. (See FP 14.28.)
- ☐ is not recognized as an officer of the assignee. (See FP 14.29.)
- ☒ does not have power of attorney, and thus, is not of record. (See FP 14.29.01.)

(Note: PoA can be given to a customer number, wherein all practitioners listed under the customer number have PoA. If PoA is established by a list of practitioners, the list may not comprise more than 10 practitioners. A representative of the assignee, who is not of record, cannot sign the TD unless it is established that the representative is a party authorized to act on behalf of the assignee.)

☐ The TD is not supported by evidence of chain of title to the assignee signing the TD due to a failure to submit either: (a) documentary evidence of a chain of title from the original inventor(s) to the assignee and a statement affirming that the documentary evidence was, or concurrently is being, submitted for recordation; or (b) the reel and frame number(s) where such documentary evidence is recorded in the Office. 37 CFR 3.73(b). (See FPs 14.30 and 14.34)

NOTE: This documentary evidence or the specifying of the reel and frame number may be found in the TD or in a separate paper submitted by applicant.)

☐ The TD is not supported by adequate evidence of chain of title to the assignee signing the TD, because the person who signed the submission under 37 CFR 3.73(b):

☐ has failed to state his/her capacity to sign for the business entity. (See FPs 14.30.02 and 14.16.02

☐ is not recognized as an officer of the assignee. (See FP 14.30.02 and 14.16.03)

(Note: On the submission under 37 CFR 3.73(b), the signature of an attorney or agent registered to practice before the Office is not sufficient, unless the attorney or agent is authorized to act on behalf of the assignee.)

☐ The TD is not signed (See FPs 14.26 and 14.26.03)

☐ The serial number of the application (or the number of the patent) which forms the basis for the double patenting is not identified (i.e., missing or incorrect) in the TD. (See FP 14.32)

☐ The serial number of the application being examined (or the number of the patent under reexam or reissue) is not identified or incorrect. (See FPs 14.26 and 14.26.04 or 14.26.05)

☐ The TD is not signed by all owners. See FPs 14.26 and 14.26.06.

☐ The period disclaimed is incorrect or not specified. (See FPs 14.24, 14.27.02 or 14.27.03)

☐ Other

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on May 17, 2011.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action dated February 18, 2011, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.111.

Claims 1-25 remain pending in the application. Claims 1, 2, 10, 11, 18, and 19 have been amended. No claims have been added or canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 8.

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions of claims in the application.

1. (Currently amended) A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising:

decoding from the compressed video sequence an indication related to ~~at least one~~an intentional discontinuity of numbering of the image frames;

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to [[the]]at least one discontinuity in the numbering of the image frames; and

using the image frames in the buffer memory in the decoding process.

2. (Currently amended) A method according to claim 1, wherein ~~said indication informs that~~ the at least one discontinuity in the numbering of the image frames in the video sequence is intentional.

3. (Previously presented) A method according to claim 1, further comprising

using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

4. (Original) A method according to claim 1, wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.

5. (Previously presented) A method according to claim 1, further comprising

continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

6. (Previously presented) A method according to claim 1, further comprising

entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

7. (Previously presented) A method according to claim 6, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.

8. (Previously presented) A method according to claim 1, wherein

configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.

9. (Previously presented) A method according to claim 3, further comprising

detecting an error in the compressed video sequence on a basis of a reference in the decoding process to an image frame corresponding to the at least one discontinuity in the numbering of image frames.

10. (Currently amended) A video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding, the video decoder being configured to

decode from the compressed video sequence an indication related to ~~at least one~~an intentional discontinuity of numbering of the image frames;

configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to [[the]]at least one discontinuity in the numbering of the image frames; and

use the image frames in the buffer memory in the decoding process.

11. (Currently amended) A video decoder according to claim 10, wherein

~~said indication informs that~~ the at least one discontinuity in the numbering of the image frames in the video sequence is intentional.

12. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

use the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

13. (Original) A video decoder according to claim 10, wherein

the number of image frames in the buffer memory are used in sliding window buffering of image frames.

14. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

continue the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

15. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

enter into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

16. (Previously presented) A video decoder according to claim 15, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.

17. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

configure the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

18. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, image frames of said video sequence being entered into a buffer memory in connection with the decoding, the computer program product comprising:

a computer program code for decoding from the compressed video sequence an indication related to ~~at least one~~an intentional discontinuity of numbering of the image frames;

a computer program code for configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to [[the]]at least one discontinuity in the numbering of the image frames; and

a computer program code for including the image frames in the buffer memory in the decoding process.

19. (Currently amended) A computer program product according to claim 18, wherein ~~said indication informs that~~ the at least one discontinuity in the numbering of the image frames in the video sequence is intentional.

20. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for including the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

21. (Original) A computer program product according to claim 18, the computer program product further comprising

a computer program code for using the number of image frames in the buffer memory in sliding window buffering of image frames.

22. (Previously presented) A computer program product according to claim 18, the computer program product further comprising

a computer program code for continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

23. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

24. (Previously presented) A computer program product according to claim 23, the computer program product further comprising:

a computer program code for designating said filler frames by an identifier indicating that the filler frames do not belong to the compressed video sequence.

25. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

26-31. (Canceled)

REMARKS/ARGUMENTS

The reference to "Transitional After Final Practice" on page two of the Office action appears to be in error. Applicant notes that this application does not meet the timing requirements for Rule 1.129(a) and no such fee was paid. Applicant requests that these comments be withheld from further actions or that an explanation be provided.

The rejections presented in the Office action dated February 18, 2011 (hereinafter "Office action") have been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

With particular respect to claims 3, 5-7, 9, 10, 12, 14-16, 18, and 20-25, Applicant notes that the claims are not included in any statement of rejection. MPEP § 707.07(d) indicates that where a claim is refused the word "reject" must be used and the statutory basis for any ground of rejection should be designated by an express reference in the opening sentence of each ground of rejection. While it appears that these claims were intended to be rejected on the same grounds as respective claims 1 and 18, these claims are believed to be allowable for the reasons set forth below in connection with claims 1 and 18. If this was not the Examiner's intention, Applicant requests clarification, an opportunity to respond, and that any future rejections comply with MPEP § 707.07(d).

Applicant appreciates the indication of allowability for claims 2, 11, and 19.

Consistent with the indication of allowable subject matter on page fourteen of the Office action, independent claims 1, 10, and 18 have been amended to include elements indicated as being directed to allowable subject matter in respective claims 2, 11, and 19. Therefore these changes do not introduce new matter. Each of the independent claims recites that an indication related to an intentional discontinuity of numbering in the image frames is decoded from the compressed video sequence. This is not recited with respect to the buffer memory configuration since the buffer memory is configured to take into account both intentional discontinuities and errors in the compressed video sequence. U.S. Patent No. 7,103,669 to Apostolopoulos (hereinafter "Apostolopoulos") fails to teach or suggest each of the now-claimed elements of the independent claims since Apostolopoulos discusses only discontinuities of numbering caused by errors. In view of these changes and the indication of

allowable subject matter, each of claims 1-25 is believed to be patentable over the asserted references.

Thus, based upon the Examiner's acknowledgement that elements now included in each of the pending claims are not found in Apostolopoulos, either alone or as modified, each of the prior art rejections would be improper. Applicant accordingly requests that each of the § 102(e) and § 103(a) rejections be withdrawn.

To the extent that the current response does not respond to any characterization in the Office action of the asserted art or of the claimed subject matter, or to any application in the Office action of the asserted art to any claimed subject matter, it is stated for the record that any such lack of response should not be interpreted as an acquiescence to such characterizations or applications. A detailed discussion of each of the Office action's characterizations, or any other assertions or statements beyond that provided above is unnecessary in view of the present response. The right to address in detail any such assertions or statements in the future is reserved.

Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C1) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the Examiner is invited to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.232.6907

Date: May 17, 2011

By: /Erin Nichols Matkaiti/

Erin Nichols Matkaiti
Reg. No. 57,125

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. § 1.97(c)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant respectfully submits the items of information on the enclosed Form 1449 for the attention of the Examiner in the above-identified application.

This statement should be considered because it is submitted after the mailing date of the first Office action on the merits but before the mailing date of a final Office action under 37 C.F.R. § 1.113 or a Notice of Allowance under 37 C.F.R. § 1.311. Please charge Deposit Account 50-3581 (TPO.004.C1) the amount of $180.00 under 37 C.F.R. § 1.17(p) for consideration of the items.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. § 1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.232.6907

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti
Reg. No. 57,125

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5,680,322	10/21/1997	Shinoda			
	5,786,858	07/28/1998	Yagasaki et al.			
	6,483,875	11/19/2002	Hasebe et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	03/04/2011	Office action dated 3/4/2011 from U.S. Serial No. 11/338,934, 14 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,934	01/25/2006	Miska Hannuksela	TPO.004.C2	6610

10888 7590 03/04/2011

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

NOTIFICATION DATE	DELIVERY MODE
03/04/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

tdotter@hfiplaw.com
roswood@hfiplaw.com

Office Action Summary	Application No.	Applicant(s)	
	11/338,934	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	SIU M. LEE	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 February 2011*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-6,8-11,13 and 14* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-6,8-11,13 and 14* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *1/25/2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☒ Certified copies of the priority documents have been received in Application No. 10/306942.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Transitional After Final Practice

1. Since this application is eligible for the transitional procedure of 37 CFR 1.129(a), and the fee set forth in 37 CFR 1.17(r) has been timely paid, the finality of the previous Office action is hereby withdrawn pursuant to 37 CFR 1.129(a). Applicant's first submission after final filed on 12/3/2011 has been entered.

Response to Arguments

2. Applicant's arguments, see page 8-9, filed on 12/3/2011, with respect to the rejections of claims 5-8 and 13 under 35 U.S.C. § 101 and claims 5-8 and 13 under 35 U.S.C. § 112, 1st paragraph have been fully considered and are persuasive. Therefore, the rejection has been withdrawn.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1-2, 4-6, 8-11, 13-14 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Independent claims 1, 5, 9-10, 13-14 all recite in the preamble a limitation "at least one reference image frame is predictable from at least one previous image frame,

and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order". As the limitation states, the reference image frame is predictable from a previous image frame, therefore, it is essential for the previous image frame to be decoded before it can use to predict the reference image frame. It is unclear when the reference image frame is decoded earlier than the previous image frame, it would not be predict from the previous image frame since the reference image frame is earlier in decoding order. As stated in the abstract of the instant application, "In the decoding phase, the indication of at least one image frame is decoded from the video sequence, and the decoding of the video sequence is started from **said first image frame of the independent sequence**, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame"; that is the first I frame of the sequence is always being decode first. Therefore, it is unclear what "at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order" means.

Independent claims 9-10, 13-14 all recite in the preamble a limitation "resetting the identifier value for the indicated first image frame of the independent sequence"; as disclose in paragraph 0018 of the publication of the instant application (2006/0120451 A1), the resetting step of the identifier value for the indicated first picture of an independent sequence is performed in the encoder, however, claims 9-11, 13-14 are

directed to a decoding, it is unclear how the decoder reset the identifier value of the indicated first image frame of the independent sequence.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2).

(1) Regarding claim 1:

(claim 1 is rejected without the consideration of the limitation in the preamble because of the 112 rejection as explained in above)

Apostolopoulos discloses an encoder comprising:

encoding into the video sequence an indication of at least one image frame, which is the first image frame (encoded I1 frame as shown in figure 4), in decoding order, of the independent sequence (decoding of the P frames are depending on the I1 frame), said at least one reference image frame being included in the sequence (all P frames are predict from I1 frame, the examiner interprets the claimed at least one reference image frame as one of the P frames as shown in figure 4);

encoding identifier values for the image frames according to a numbering scheme (the P frames are decoded sequentially from I1, column 9, lines 60-65; the numbering scheme of odd 1,3,5….15 and even 2,4,6…).

Apostolopoulos fails to explicitly disclose resetting the identifier value for the indicated first image frame of the independent sequence.

However, each sequence that is encoded is independent and it would have been obvious to one of ordinary skill in the art that it is not necessary to incrementing the identifier of the first image frame because it prevent using extra bit to represent a higher number of identifier and reduce the amount of data to be transmitted to the decoder.

(2) Regarding claim 10:

(claim 10 is rejected without the consideration of the limitation in the preamble because of the 112 rejection as explained in above, the examiner interprets the resetting of the identifier values are reset in the encoder)

Apostolopoulos discloses a method for decoding comprising:

decoding from the video sequence an indication of at least one image frame, which is the first image frame (the decoder decode the even frames and odd frames sequence, column 8, lines 40-49 as disclose in figure 4, the first frame is identified as an I1 frame for prediction of the following frames, column 9, lines 60-65), in decoding order, of an independent sequence, said sequence including the at least one reference image frame (all P frames are predict from I1 frame, the examiner interprets the claimed at least one reference image frame as one of the P frames as shown in figure 4);

starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame (all the other P frames are depend on the I1 frame, therefore, no prediction is need from any image frame prior to I1, column 9, lines 60-65);

decoding identifier values for image frames according to a numbering scheme (the P frames are decoded sequentially from I1, column 9, lines 60-65).

Apostolopoulos fails to explicitly disclose resetting the identifier value for the indicated first image frame of the independent sequence.

However, each sequence that is encoded is independent and it would have been obvious to one of ordinary skill in the art that it is not necessary to incrementing the identifier of the first image frame because it prevent using extra bit to represent a higher number of identifier and reduce the amount of data to be transmitted to the decoder.

7. Claims 2 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) as applied to claims 1 and 10 above, and further in view of Yagasaki et al. (US 5,786,858).

Apostolopoulos discloses all subject matter as discussed in claim 10 except wherein the indication is a separate flag included in the header of a slice.

However, Yagasaki discloses putting a flag for indicating an intra-frame coded block in the slice header (column 10, line 65 – column 11, line 2).

It is desirable to have the indication is a separate flag included in the header of a slice because it prevents a decoding process of the entire bit stream for extraction of the flag (column 11, lines 1-2). Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the teaching of Yagasaki in the decoder of Apostolopoulos to simplify the extraction process of the identifier.

8. Claims 5, 9, and 13-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) as applied to claim 10 above, and further in view of Langberg et al. (US 5,852,630).

(1) Regarding claims 5 and 9:

Apostolopoulos discloses an encoder comprising:

encoding into the video sequence an indication of at least one image frame, which is the first image frame (encoded I1 frame as shown in figure 4), in decoding order, of the independent sequence (decoding of the P frames are depending on the I1 frame), said at least one reference image frame being included in the sequence (all P frames are predict from I1 frame, the examiner interprets the claimed at least one reference image frame as one of the P frames as shown in figure 4);

encoding identifier values for the image frames according to a numbering scheme (the P frames are decoded sequentially from I1, column 9, lines 60-65; the numbering scheme of odd 1,3,5….15 and even 2,4,6…).

Apostolopoulos fails to explicitly disclose (a) resetting the identifier value for the indicated first image frame of the independent sequence; and (b) a computer program

product stored on a non-transitory computer readable medium and executable in a data processing device for implementing the method.

With respect to (a), each sequence that is encoded is independent and it would have been obvious to one of ordinary skill in the art that it is not necessary to incrementing the identifier of the first image frame because it prevent using extra bit to represent a higher number of identifier and reduce the amount of data to be transmitted to the decoder.

With respect to (b), Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system (processor) or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Apostolopoulos would have been implemented in a software. The implemented software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg et al. for implementing the decoding method of Apostolopoulos in order to reduce cost and improve the adaptability and flexibility of the encoder.

(2) Regarding claim 13 and 14:

Apostolopoulos discloses a method for decoding comprising:

decoding from the video sequence an indication of at least one image frame, which is the first image frame (the decoder decode the even frames and odd frames sequence, column 8, lines 40-49 as disclose in figure 4, the first frame is identified as an I1 frame for prediction of the following frames, column 9, lines 60-65), in decoding order, of an independent sequence, said sequence including the at least one reference image frame (all P frames are predict from I1 frame, the examiner interprets the claimed at least one reference image frame as one of the P frames as shown in figure 4);

starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame (all the other P frames are depend on the I1 frame, therefore, no prediction is need from any image frame prior to I1, column 9, lines 60-65);

decoding identifier values for image frames according to a numbering scheme (the P frames are decoded sequentially from I1, column 9, lines 60-65).

Apostolopoulos fails to explicitly disclose (a) resetting the identifier value for the indicated first image frame of the independent sequence; and (b) the decoder comprising at least one processor and at leas tone memory including computer program code to perform the method.

With respect to (a), each sequence that is encoded is independent and it would have been obvious to one of ordinary skill in the art that it is not necessary to incrementing the identifier of the first image frame because it prevent using extra bit to

represent a higher number of identifier and reduce the amount of data to be transmitted to the decoder.

with respect to (b), Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system (processor) or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Apostolopoulos would have been implemented in a software. The implemented software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg et al. for implementing the decoding method of Apostolopoulos in order to reduce cost and improve the adaptability and flexibility of the decoder.

9. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) as applied to claim 10 above, and further in view of Yagasaki et al. (US 5,786,858).

Apostolopoulos discloses all subject matter as discussed in claim 10 except wherein the indication is a separate flag included in the header of a slice.

However, Yagasaki discloses putting a flag for indicating an intra-frame coded block in the slice header (column 10, line 65 – column 11, line 2).

It is desirable to have the indication is a separate flag included in the header of a slice because it prevents a decoding process of the entire bit stream for extraction of the flag (column 11, lines 1-2). Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the teaching of Yagasaki in the decoder of Apostolopoulos to simplify the extraction process of the identifier.

10. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) in view of Langberg et al. (US 5,852,630) as applied to claim 5 above, and further in view of Yagasaki et al. (US 5,786,858).

Apostolopoulos and Langberg discloses all subject matter as discussed in claim 5 except wherein the indication is a separate flag included in the header of a slice.

However, Yagasaki discloses putting a flag for indicating an intra-frame coded block in the slice header (column 10, line 65 – column 11, line 2).

It is desirable to have the indication is a separate flag included in the header of a slice because it prevents a decoding process of the entire bit stream for extraction of the flag (column 11, lines 1-2). Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the teaching of Yagasaki in the decoder of Apostolopoulos and Langberg to simplify the extraction process of the identifier.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Hasebe et al. (US 6,483,875 B1) discloses a picture signal processing apparatus.

Shinoda (US 5,680,322) discloses a method and apparatus for dynamic image data transmission.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU M. LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 7:30-4:00 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Examiner, Art Unit 2611
2/28/2011

Notice of References Cited	Application/Control No. 11/338,934	Applicant(s)/Patent Under Reexamination HANNUKSELA, MISKA	
	Examiner SIU M. LEE	Art Unit 2611	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-5,680,322	10-1997	Shinoda, Mayumi	714/18
*	B	US-6,483,875	11-2002	Hasebe et al.	375/240.15
*	C	US-5,786,858	07-1998	Yagasaki et al.	375/240.15
*	D	US-7,103,669	09-2006	Apostolopoulos, John G.	709/231
*	E	US-5,852,630	12-1998	Langberg et al.	375/219
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Electronic Patent Application Fee Transmittal

Application Number:	11338996
Filing Date:	25-Jan-2006
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Filer:	Erin M. Nichols
Attorney Docket Number:	TPO.004.C1

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
		Total in USD ($)		**180**

Electronic Acknowledgement Receipt

EFS ID:	10108026
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	10888
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	17-MAY-2011
Filing Date:	25-JAN-2006
Time Stamp:	14:59:17
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$180
RAM confirmation Number	1296
Deposit Account	503581
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		OAR-TPO004C1-OA18022011.pdf	79696 47300f709550cb902f658bd15bd927096f7e4390	yes	9

Multipart Description/PDF files in .zip description

Document Description	Start	End
Amendment/Req. Reconsideration-After Non-Final Reject	1	1
Claims	2	7
Applicant Arguments/Remarks Made in an Amendment	8	9

Warnings:

Information:

2	Transmittal Letter	IDS-14-TPO004C1.pdf	55999 b8e9100de5245e62217a040a96728032963 2a635	no	2

Warnings:

Information:

3	Information Disclosure Statement (IDS) Filed (SB/08)	1449-14-TPO004C1.pdf	58530 f536f21483f4f7bccea9f709fc22c0aa0cf9b6 4b	no	1

Warnings:

Information:

This is not an USPTO supplied IDS fillable form

4	NPL Documents	OA-04032011.pdf	494640 20672be921c11d4ee09997c4dabd885afaa 8287f	no	14

Warnings:

Information:

5	Fee Worksheet (PTO-875)	fee-info.pdf	30216 300df5b9481c81d957fae1ce69d9f078657f 9695	no	2

Warnings:

Information:

Total Files Size (in bytes):	719081

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
11/338,996	01/25/2006	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY ☐ RATE ($)	SMALL ENTITY ☐ FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	05/17/2011	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 25	Minus	** 31	= 0	X $ =		OR	X $52=	0
	Independent (37 CFR 1.16(h))	* 3	Minus	***6	= 0	X $ =		OR	X $220=	0
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/TINA J. BARDEN/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,996	01/25/2006	Miska Hannuksela	TPO.004.C1	5361

10888 7590 08/09/2011

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

NOTIFICATION DATE	DELIVERY MODE
08/09/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

tdotter@hfiplaw.com
roswood@hfiplaw.com

Office Action Summary	Application No.	Applicant(s)
	11/338,996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU LEE	2611

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 May 2011*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *1/25/2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☒ Certified copies of the priority documents have been received in Application No. 10/306,942.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *4/5/2011, 5/17/2011*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments with respect to claims 1-25 have been considered but are moot in view of the new ground(s) of rejection based on new interpretation of Apostolopoulos (US 7,103,669 B2).

Claim Objections

2. Claims 2, 11, and 19 are objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of previous claims 1, 10, and 18 respectively. Applicant is required to cancel the claims, or amend the claims to place the claims in proper dependent form, or rewrite the claims in independent form.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1-3, 5-7, 9-12, 14-16 are rejected under 35 U.S.C. 102(e) as being anticipated by Apostolopoulos (US 7,103,669 B2).

(1) Regarding claim 1:

Apostolopoulos disclose a method for decoding a compressed video sequence (odd number sequence ort even number sequence), wherein image frames are entered into a buffer memory in connection with the decoding (decode 322 in figure 5 enter the decoded frame into the frame memory including the previous decoded frames as shown in figure 5), the method comprising:

decoding from the compressed video sequence an indication (detected error that one of the frame (for example P5 as discussed in column 10, lines 52-63) is unable to be retrieved due to packet loss) related to an intentional discontinuity of numbering of the image frame (the packet is decoded in step 914 and a determination is made whether an error has been detected, when an error is detected, state recovery is performed by employing previous and future frames of correctly decoded frames, column 7, lines 9-37; the examiner interprets the detected error is related to the intentional discontinuity (frames in subsequence of the even frames that is missing from the subsequence of the odd frames) since the error rate is reconstruct by interpolation or estimation based on the odd frame);

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to the art least one discontinuity in the numbering of the image frames (the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18); and

using the image frames in the buffer memory in the decoding process (the examiner interprets the process within the dot line as shown in figure 9 as the decoding process, as the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18, therefore, full frame versions of P2, P4, and P6 through P14, as well we P1 and P3 in the buffer memory are use in the decoding process).

(2) Regarding claim 10:

Apostolopoulos discloses a video decoder (multiple state video decoding 128 in figure 1 comprises decode block 320 and 322 as shown in figure 5) for decoding a compressed video sequence, the video decoder being connected to a buffer memory (frame memory as shown in figure 5) wherein image frames are entered in connection with the decoding, the video decoder being configured to

decode from the compressed video sequence an indication (detected error that one of the frame (for example P5 as discussed in column 10, lines 52-63) is unable to be retrieved due to packet loss) related to an intentional discontinuity of numbering of the image frames (the packet is decoded in step 914 and a determination is made whether an error has been detected, when an error is detected, state recovery is performed by employing previous and future frames of correctly decoded frames, column 7, lines 9-37; the examiner interprets the detected error is related to the intentional discontinuity (frames in subsequence of the even frames that is missing from the subsequence of the odd frames)

since the error rate is reconstruct by interpolation or estimation based on the odd frame); and

use the image frames in the buffer memory in the decoding process (the examiner interprets the process within the dot line as shown in figure 9 as the decoding process, as the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18, therefore, full frame versions of P2, P4, and P6 through P14, as well we P1 and P3 in the buffer memory are use in the decoding process).

(3) Regarding claims 3 and 12:

Apostolopoulos further disclose using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; the examiner interprets “do not exist at a decoder due to the at least one discontinuity in the numbering of the image frame" as due to the error occur in the transmission of the encoded odd frame and prevent the decode block from properly reconstructing P_5, therefore there is a discontinuity in the numbering of image frame, column 9, lines 66-67).

(4) Regarding claim 9:

Apostolopoulos further disclose detecting an error in the compressed video sequence on a basis of a reference in the decoding process to an image

frame corresponding to the at least one discontinuity in the numbering of image frames (the error occur in the transmission of the encoded odd frame and prevent the decode block from properly reconstructing P_5, therefore there is a discontinuity in the numbering of image frame, column 9, lines 66-67).

(5) Regarding claims 5 and 14:

Apostolopoulos further disclose continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames (the receiver 332 includes a state recovery 526 for recovering the lost P-frame P5 of the stream 354 using the information provided by the stream 356, the recovery of P5 enables the decode block 322 to continue with its prediction loop by correctly updating its state information and reconstruct p7 to P13, column 10, lines 55-60; as the frames are correctly decode by the decoder, frames are enter into the frame memory as a previous decoded frames as shown in figure 5).

(6) Regarding claims 6 and 15:

Apostolopoulos further discloses entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; the examiner interprets the correctly decode frame that replace the lost frame as the claimed “filler frame”).

(7) Regarding claims 7 and 16:

Apostolopoulos further discloses wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; therefore an odd frame can be replace by an even frame, wherein the even frame number (identifier) indicates that the filler frame do not belong to the odd frame sequence).

(8) Regarding claim 2 and 11:

Apostolopoulos further discloses the at least one discontinuity in the numbering of the image frames in the video sequence is intentional (the discontinuity in the odd frame sequence (even frame is missing) is intentional for the purpose of path diversity, therefore, it is intentional, abstract).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 18-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) in view of Langberg (US 5,852,630).

(1) Regarding claim 18:

Apostolopoulos disclose a method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding (decode 322 in figure 5 enter the decoded frame into the frame memory including the previous decoded frames as shown in figure 5), the method comprising:

decoding from the compressed video sequence an indication (detected error that one of the frame (for example P5 as discussed in column 10, lines 52-63) is unable to be retrieved due to packet loss) related to an intentional discontinuity of numbering of the image frame (the packet is decoded in step 914 and a determination is made whether an error has been detected, when an error is detected, state recovery is performed by employing previous and future frames of correctly decoded frames, column 7, lines 9-37; the examiner interprets the detected error is related to the intentional discontinuity (frames in subsequence of the even frames that is missing from the subsequence of the odd frames) since the error rate is reconstruct by interpolation or estimation based on the odd frame); and

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to the art least one discontinuity in the numbering of the image frames (the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18); and

using the image frames in the buffer memory in the decoding process (the examiner interprets the process within the dot line as shown in figure 9 as the decoding process, as the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18, therefore, full frame versions of P2, P4, and P6 through P14, as well we P1 and P3 in the buffer memory are use in the decoding process).

Apostolopoulos discloses all of the subject matter as described above except for the method is perform by a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device.

However, Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Apostolopoulos would have been implemented in a software. The implemented software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg for

implementing the method of Apostolopoulos in order to reduce cost and improve the adaptability and flexibility of the communication system.

(2) Regarding claim 20:

Apostolopoulos further disclose using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; the examiner interprets “do not exist at a decoder due to the at least one discontinuity in the numbering of the image frame" as due to the error occur in the transmission of the encoded odd frame and prevent the decode block from properly reconstructing P_5, therefore there is a discontinuity in the numbering of image frame, column 9, lines 66-67).

(3) Regarding claim 21:

Apostolopoulos disclose all subject matter as discussed in claim 1 but fails to explicitly disclose wherein the number of image frames in the buffer memory is used in sliding widow buffering of image frames.

However, as shown in figure 5, decoded frame as input into the frame memory as previous decode frames, these frames are used for recovery of a lost frame (recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstructed full frame versions of P2, P4, P6 through P14 as well as P1 and P3, column 11, lines 14-18), therefore, the frame memory have to store the decoded frame P2 to P14 in the frame memory on the even frame number sequence and obviously, the frame

memory have to store the decoded frame P1 to P13 in the frame memory on the odd frame number sequence, therefore, the examiner interprets the frame memory are used in sliding window buffering of the decode image frame.

It is desirable to have the number of image frames in the buffer memory are used in sliding widow buffering of image frames because the buffer of image frame is need for recovering of a full frame version of a lost frame and provide the benefit of increase the image quality of the recovered video sequence.

(4) Regarding claim 22:

Apostolopoulos further disclose continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames (the receiver 332 includes a state recovery 526 for recovering the lost P-frame P5 of the stream 354 using the information provided by the stream 356, the recovery of P5 enables the decode block 322 to continue with its prediction loop by correctly updating its state information and reconstruct p7 to P13, column 10, lines 55-60; as the frames are correctly decode by the decoder, frames are enter into the frame memory as a previous decoded frames as shown in figure 5; the number of image frames corresponding to the at least one discontinuity are the correctly decoded even frames).

(5) Regarding claim 23:

Apostolopoulos further discloses entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; the examiner interprets the correctly decode frame that replace the lost frame as the claimed "filler frame").

(6) Regarding claim 24:

Apostolopoulos further discloses wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence (the lost frame may be estimated by replacing it by another correctly decoded frame, column 11, lines 48-49; therefore an odd frame can be replace by an even frame, wherein the even frame number (identifier) indicates that the filler frame do not belong to the odd frame sequence).

(7) Regarding claim 25:

Apostolopoulos further discloses wherein configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by frame number, whereby no data is entered into the buffer memory (past and future frames with respect to the lost frame P5 to recover frame P5, column 10, lines 60-63, as the past and future frames are identified by frame number and when the decoder cannot decode P5, the decode block 322 cannot reconstruct P5 to P13, therefore, no data is entered into the buffer memory (frame memory that store previous decoded frames)).

Apostolopoulos fails to disclose using memory indications for reconstruction of a lost frame.

However, as shown in figure 5, the correctly decoded frames are storage in the frame memory as previous decoded frames. Therefore, it would have been obvious to one of ordinary skill in the art to modify the invention Apostolopoulos to use memory indication (location) for locating future frames in the even frame number sequence and past frame in the odd frame number sequence for reconstruction of the lose frame since using the frame number and memory location are interchangeable to one of ordinary skill in the art.

(8) Regarding claim 19:

Apostolopoulos further discloses the at least one discontinuity in the numbering of the image frames in the video sequence is intentional (the discontinuity in the odd frame sequence (even frame is missing) is intentional for the purpose of path diversity, therefore, it is intentional, abstract).

7. Claims 4, 8, 13, and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2).

(1) Regarding claim 4 and 13:

Apostolopoulos disclose all subject matter as discussed in claim 1 but fails to explicitly disclose wherein the number of image frames in the buffer memory is used in sliding widow buffering of image frames.

However, as shown in figure 5, decoded frame as input into the frame memory as previous decode frames, these frames are used for recovery of a lost

frame (recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstructed full frame versions of P2, P4, P6 through P14 as well as P1 and P3, column 11, lines 14-18), therefore, the frame memory have to store the decoded frame P2 to P14 in the frame memory on the even frame number sequence and obviously, the frame memory have to store the decoded frame P1 to P13 in the frame memory on the odd frame number sequence, therefore, the examiner interprets the frame memory are used in sliding window buffering of the decode image frame.

It is desirable to have the number of image frames in the buffer memory are used in sliding widow buffering of image frames because the buffer of image frame is need for recovering of a full frame version of a lost frame and provide the benefit of increase the image quality of the recovered video sequence.

(2) Regarding claims 8 and 17:

Apostolopoulos further discloses wherein configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by frame number, whereby no data is entered into the buffer memory (past and future frames with respect to the lost frame P5 to recover frame P5, column 10, lines 60-63, as the past and future frames are identified by frame number and when the decoder cannot decode P5, the decode block 322 cannot reconstruct P5 to P13, therefore, no data is entered into the buffer memory (frame memory that store previous decoded frames)).

Apostolopoulos fails to disclose using memory indications for reconstruction of a lost frame.

However, as shown in figure 5, the correctly decoded frames are storage in the frame memory as previous decoded frames. Therefore, it would have been obvious to one of ordinary skill in the art to modify the invention Apostolopoulos to use memory indication (location) for locating future frames in the even frame number sequence and past frame in the odd frame number sequence for reconstruction of the lose frame since using the frame number and memory location are interchangeable to one of ordinary skill in the art.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 6:15-3:45 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Examiner, Art Unit 2611
7/26/2011

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

SEARCHED

Class	Subclass	Date	Examiner
375	240.27, 240.25	6/14/2010	Siu M. Lee

SEARCH NOTES

Search Notes	Date	Examiner
EAST	6/14/2010	Siu M. Lee
Discuss the application with Chieh Fan	6/14/2010	Siu M. Lee
EAST	11/1/2010	Siu M. Lee
Consult John Peng on the 101 problem in claim 10-17	11/1/2010	Siu M. Lee
Discuss the amendment with Cheih Fan	10/27/2010	Siu M. Lee
EAST	2/14/2011	Siu M. Lee
EAST	7/25/2011	Siu M. Lee
Discuss the amendment with Chieh Fan	7/25/2011	Siu M. Lee

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	06/13/2010	11/01/2010	02/14/2011	07/26/2011					
	1	✓	✓	✓	✓					
	2	✓	✓	O	✓					
	3	✓	✓	✓	✓					
	4	✓	✓	✓	✓					
	5	✓	✓	✓	✓					
	6	✓	✓	✓	✓					
	7	O	O	✓	✓					
	8	O	O	✓	✓					
	9	O	✓	✓	✓					
	10	✓	✓	✓	✓					
	11	✓	✓	O	✓					
	12	✓	✓	✓	✓					
	13	✓	✓	✓	✓					
	14	✓	✓	✓	✓					
	15	✓	✓	✓	✓					
	16	✓	O	✓	✓					
	17	✓	O	✓	✓					
	18	✓	✓	✓	✓					
	19	✓	✓	O	✓					
	20	✓	✓	✓	✓					
	21	✓	✓	✓	✓					
	22	✓	✓	✓	✓					
	23	✓	✓	✓	✓					
	24	✓	O	✓	✓					
	25	✓	O	✓	✓					
	26	✓	-	-	-					
	27	✓	-	-	-					
	28	✓	-	-	-					
	29	✓	-	-	-					
	30	✓	-	-	-					
	31	✓	-	-	-					

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 5361

SERIAL NUMBER	FILING or 371(c) DATE / RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/338,996	01/25/2006	375	2611	TPO.004.C1

APPLICANTS
Miska Hannuksela, Tampere, FINLAND;

**** CONTINUING DATA ***************************
This application is a CON of 10/306,942 11/29/2002 PAT 7,894,521

**** FOREIGN APPLICATIONS ***************************
FINLAND 20020127 01/23/2002

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****
02/28/2006

Foreign Priority claimed ☑ Yes ☐ No 35 USC 119(a-d) conditions met ☑ Yes ☐ No Verified and Acknowledged /SIU M LEE/ Examiner's Signature	☐ Met after Allowance Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		FINLAND	4	31	6

ADDRESS
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

TITLE
Grouping of image frames in video coding

FILING FEE RECEIVED	FEES: Authority has been given in Paper	☐ All Fees
2150	No._____ to charge/credit DEPOSIT ACCOUNT	☐ 1.16 Fees (Filing)
	No._____ for following:	☐ 1.17 Fees (Processing Ext. of time)
		☐ 1.18 Fees (Issue)
		☐ Other _____
		☐ Credit

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/SL/	5,680,322	10/21/1997	Shinoda			
/SL/	5,786,858	07/28/1998	Yagasaki et al.			
/SL/	6,483,875	11/19/2002	Hasebe et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	03/04/2011	Office action dated 3/4/2011 from U.S. Serial No. 11/338,934, 14 pages.

EXAMINER	/Siu Lee/	DATE CONSIDERED	07/05/2011

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	2002	Hannuksela, "Signaling of Enhanced GOPs", Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 4th Meeting, Austria, July 22-26, 2002.
/SL/	4/5/2011	File History for EP Application No. 03700321.7 as retrieved from the European Patent Office Electronic File System on 4/5/2011, 222 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 07/05/2011

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2	"5680322".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/26 14:16
L2	2	"5786858".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/26 14:37
L3	2	"6483875".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/26 14:38
S1	18667	video compression	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/24 18:39
S2	2714	S1 and ((tag or number or indicat$3 or frame) with (omit$4 or discontinu$4 or skip$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/24 18:41
S3	1368	S2 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/24 18:41
S4	86	S3 and ((reconstruct$3) with (omit$4 or discontinu$4 or skip$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/24 18:42

EAST Search History (Interference)

<This search history is empty>

7/26/2011 2:52:46 PM
C:\Users\slee7\Documents\EAST\Workspaces\11338996 201107.wsp

To: tdotter@hfiplaw.com,roswood@hfiplaw.com,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 10888

Aug 09, 2011 05:22:55 AM

Dear PAIR Customer:

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 10888 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
11338996	CTNF	08/09/2011	TPO.004.C1
	1449	08/09/2011	TPO.004.C1
	1449	08/09/2011	TPO.004.C1

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	JP 8265694	10/11/1996	JP			Abstract	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	08/02/2011	Notice of Allowance dated 8/2/2011 from Japanese Application No. 2006-120296, 3 pages.
	08/19/2011	File History for U.S. Serial No.11/338,934 as retrieved from the U.S. Patent and Trademark Office PAIR System on 8/19/2011, 246 pages.

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

10888 7590 08/18/2011

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER
LEE, SIU M

ART UNIT	PAPER NUMBER
2611	

DATE MAILED: 08/18/2011

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,934	01/25/2006	Miska Hannuksela	TPO.004.C2	6610

TITLE OF INVENTION: GROUPING OF IMAGE FRAMES IN VIDEO CODING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	11/18/2011

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

10888 7590 08/18/2011
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,934	01/25/2006	Miska Hannuksela	TPO.004.C2	6610

TITLE OF INVENTION: GROUPING OF IMAGE FRAMES IN VIDEO CODING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	11/18/2011

EXAMINER	ART UNIT	CLASS-SUBCLASS
LEE, SIU M	2611	375-240120

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:
- ❑ Issue Fee
- ❑ Publication Fee (No small entity discount permitted)
- ❑ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ❑ A check is enclosed.
- ❑ Payment by credit card. Form PTO-2038 is attached.
- ❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ❑ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ❑ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,934	01/25/2006	Miska Hannuksela	TPO.004.C2	6610

10888 7590 08/18/2011

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

DATE MAILED: 08/18/2011

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 1107 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 1107 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No.	Applicant(s)
	11/338,934	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU LEE	2611

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *amendment filed on 6/2/2011*.
2. ☒ The allowed claim(s) is/are *1,2,4-6,8-11,13 and 14*.
3. ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☒ All b) ☐ Some* c) ☐ None of the:
      1. ☒ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)	5. ☐ Notice of Informal Patent Application
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)	6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
3. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date 4/5/2011, 6/2/2011	7. ☐ Examiner's Amendment/Comment
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material	8. ☒ Examiner's Statement of Reasons for Allowance
	9. ☐ Other _____.

DETAILED ACTION

Allowable Subject Matter

1. Claims 1-2, 4-6, 8-11, and 13-14 are allowed.

2. The following is an examiner's statement of reasons for allowance:

(1) Regarding claims 1-2, 4-6, and 8-9:

The present invention describes a method, encoder and a computer program product for encoding a video sequence comprising an independent sequence of image frames, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence, the method comprising encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence; encoding identifier values for the image frames according to a numbering scheme; and resetting the identifier value for the indicated first image frame of the independent sequence. The closest prior art, Apostolopoulos (US 7,103,669 B2) discloses a similar method but fails to disclose encoding identifier values for the image frames according to a numbering scheme; and resetting the identifier value for the indicated first image frame of the independent sequence. This distinct feature has been added to the independent claims 1, 5, and 9, thus rendering claims 1-2, 4-6, and 8-9 allowable.

(2) Regarding claims 10-11, 13-14:

The present invention describes a method, decoder, and a computer program product for decoding a compressed video sequence, comprising decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence; starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame; decoding identifier values for image frames according to a numbering scheme; and resetting the identifier value for the indicated first image frame of the independent sequence. The closest prior art, Apostolopoulos (US 7,103,669 B2) discloses a similar method but fails to disclose starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame; decoding identifier values for image frames according to a numbering scheme; and resetting the identifier value for the indicated first image frame of the independent sequence. This distinct feature has been added to independent claims 10, 13 and 14, thus rendering claims 10-11 and 13-14 allowable.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably

accompany the issue fee. Such submissions should be clearly labeled “Comments on Statement of Reasons for Allowance.”

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 6:15-3:45 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Examiner, Art Unit 2611
8/12/2011
/Shuwang Liu/
Supervisory Patent Examiner, Art Unit 2611

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	2002	Hannuksela, "Signaling of Enhanced GOPs", Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 4th Meeting, Austria, July 22-26, 2002.
/SL/	4/5/2011	File History for EP Application No. 03700321.7 as retrieved from the European Patent Office Electronic File System on 4/5/2011, 222 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 08/12/2011

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	2/18/2011	Office Action dated 2/18/2011 from U.S. Serial No. 11/338,996, 18 pages.
/SL/	5/17/2011	Office Action Response dated 5/17/2011 from U.S. Serial No. 11/338,996, 9 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 08/12/2011

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,934	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on June 2, 2011.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action mailed March 4, 2011, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.111.

Claims 1, 2, 4-6, 8-11, 13, and 14 remain pending in the application. Claims 1, 5, 9, 10, 13, and 14 have been amended. No claims have been added or canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 7.

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions of claims in the application.

1. (Currently amended) A method for encoding a video sequence comprising an independent sequence of image frames, wherein ~~at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order~~all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence, the method comprising:

encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence~~, said at least one reference image frame being included in the sequence~~;

encoding identifier values for the image frames according to a numbering scheme; and

resetting the identifier value for the indicated first image frame of the independent sequence.

2. (Original) A method according to claim 1, further comprising:

encoding the indication into the video sequence as a separate flag included in the header of a slice.

3. (Canceled)

4. (Original) A method according to claim 1, further comprising:

encoding into the video sequence an identifier value for the independent sequence.

5. (Currently amended) A video encoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video encoder to

encode a video sequence comprising an independent sequence of image frames, wherein ~~at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference frame is earlier than the at least one previous image frame in decoding order~~all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence by causing the video encoder to at least:

encode into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence~~, said at least one reference image frame being included in the sequence~~;

encode identifier values for the image frames according to a numbering scheme; and

reset the identifier value for the indicated first image frame of the independent sequence.

6. (Previously presented) A video encoder according to claim 5, wherein the video encoder is configured to

encode the indication into the video sequence as a separate flag included in the header of a slice.

7. (Canceled)

8. (Previously presented) A video encoder according to claim 5, wherein the video encoder is configured to

encode into the video sequence an identifier value for the independent sequence.

9. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for encoding a video sequence comprising an independent sequence of image frames, wherein ~~at least one reference image frame is predictable from at least one previous image frame and wherein the at least one reference frame is earlier than the at least one previous image frame in decoding order~~all

motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence, the computer program product comprising:

a computer program code for encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence~~, said at least one reference image frame being included in the sequence~~;

a computer code for encoding identifier values for the image frames according to a numbering scheme; and

a computer code for resetting the identifier value for the indicated first image frame of the independent sequence.

10. (Currently amended) A method for decoding a compressed video sequence, ~~wherein at least one reference image frame is predictable from at least one previous image frame and wherein the at least one reference frame is earlier than the at least one previous image frame in decoding order,~~ the method comprising:

decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, [[said]]wherein all motion-compensated temporal prediction references of the independent sequence ~~including the at least one reference~~refer only to image ~~frame~~frames within said independent sequence;

starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;

decoding identifier values for image frames according to a numbering scheme; and

resetting the identifier value for the indicated first image frame of the independent sequence.

11. (Original) A method according to claim 10, wherein the indication is a separate flag included in the header of a slice.

12. (Canceled)

13. (Currently amended) A video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence, ~~wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order~~ by causing the video decoder to at least:

decode from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, [[said]]wherein all motion-compensated temporal prediction references of the independent sequence ~~including the at least one reference~~refer only to image ~~frame~~frames within said independent sequence;

start the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame; and

decode identifier values for image frames according to a numbering scheme; and

reset the identifier value for the indicated first image frame of the independent sequence.

14. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, ~~wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order,~~ the computer program product comprising:

a computer program code for decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, [[said]]wherein all motion-compensated temporal prediction references of the

independent sequence ~~including the at least one reference~~refer only to image ~~frame~~frames within said independent sequence;

a computer program code for starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;

a computer program code for decoding identifier values for image frames according to a numbering scheme; and

a computer program code for resetting the identifier value for the indicated first image frame of the independent sequence.

15-16. (Canceled)

REMARKS/ARGUMENTS

The reference to "Transitional After Final Practice" on page two of the Office action appears to be in error. Applicant notes that this application does not meet the timing requirements for Rule 1.129(a) and no such fee was paid. Applicant requests that these comments be withheld from further actions or that an explanation be provided.

The rejections presented in the Office action dated March 4, 2011 (hereinafter "Office action") have been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

With respect to the § 112, second paragraph, rejection of the independent claims, each of the independent claims has been amended to remove the objected-to language. The language has been rewritten to characterize that "all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence". Since these changes merely re-characterize the original claim language, the changes do not introduce new matter. With particular respect to claims 9, 10, 13, and 14, Applicant notes that the identified language directed to "resetting" is located in the bodies, not the preambles, of these claims. In contrast to the assertion, claim 9 is directed to encoding. Moreover, discussion of these elements in the decoder may be found in the original specification, for example, at paragraph [0057] (using the numbering of U.S. Publication No. 2006/0120451). In view of the above, the claims particularly point out and distinctly claim the intended subject matter, and Applicant requests that the rejection be withdrawn.

Applicant respectfully traverses each of the §103(a) rejections based primarily upon the teachings of U.S. Patent No. 7,103,669 to Apostolopoulos (hereinafter "Apostolopoulos") because Apostolopoulos does not teach or suggest each of the claimed elements. For example, Apostolopoulos does not teach or suggest at least encoding into the video sequence, or decoding, an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, as claimed in each of the independent claims. Notably, Apostolopoulos does not discuss browsing a video file from a

random point and the discontinuities typically occurring in the image numbering due to such random access.

The present invention is directed to signaling to a decoder which video frame is to be decoded first as part of an independently decodable sequence of image frames, especially if there is another I-frame in the middle of the independently decodable sequence. This is discussed in paragraphs [0012-0013, 0056] (using the numbering of U.S. Publication No. 2006/0120451). Paragraph [0057] describes that there is encoded an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence (GOP), wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence, identifier values for the image frames are encoded according to a numbering scheme, and the identifier value for the indicated first image frame of the independent sequence is reset at zero.

In contrast, Apostolopoulos does not disclose or address such issues. In Apostolopoulos, a video stream is divided in two video streams, even frames encoded in a first stream and odd frame encoded in a second stream. The two streams are transmitted via separate transmission paths, decoded at reception and then combined into a complete video stream. Apostolopoulos addresses when a video file that is to be reproduced is detected to contain an error that interrupts the playback or requires the playback to be resumed from a point following the error. However, to solve this problem, Apostolopoulos suggests that if a frame is missing at the decoder, the decoding of that particular video stream is halted until the next I-frame of the video stream is received (Col. 9, line 60 – Col. 10, line 8). Thus, there is no indication, as claimed. Rather, if such an indication were included in the video stream, the decoder would immediately know where to start the decoding.

In addition, the Examiner acknowledges that Apostolopoulos does not disclose "resetting the identifier value for the indicated first image frame of the independent sequence". The assertion that such resetting would be obvious for a skilled person when aiming to prevent using an extra bit to represent higher frame numbers, is entirely unsupported. Moreover, the purpose of resetting the identifier value is not to reduce the

amount of data to be transmitted to the decoder, but to (in combination with the indication) allow the decoder to interpret the image numbers correctly and to find the correct image frame that initiates the sub-sequence from the video sequence image frames (previously cited paragraph [0057]). Without a presentation of correspondence to each of the claimed elements, each of the § 103(a) rejections is improper.

By similar reasoning, the dependent claims are also not obvious in view of the asserted references, whether each reference is considered alone or the references are considered in combination. Applicant respectfully requests withdrawal of the rejections of claims 1, 2, 4-6, 8-11, 13, and 14 and submits that all claims are in condition for allowance.

To the extent that the current response does not respond to any characterization in the Office action of the asserted art or of the claimed subject matter, or to any application in the Office action of the asserted art to any claimed subject matter, it is stated for the record that any such lack of response should not be interpreted as an acquiescence to such characterizations or applications. A detailed discussion of each of the Examiner's characterizations, or any other assertions or statements beyond that provided above is unnecessary in view of the present response. The right to address in detail any such assertions or statements in the future is reserved.

Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C2) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the Examiner is invited to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.232.6907

Date: June 2, 2011

By: /Erin Nichols Matkaiti/

Erin Nichols Matkaiti
Reg. No. 57,125

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,934	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. § 1.97(c)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant respectfully submits the items of information on the enclosed Form 1449 for the attention of the Examiner in the above-identified application.

This statement should be considered because it is submitted after the mailing date of the first Office action on the merits but before the mailing date of a final Office action under 37 C.F.R. § 1.113 or a Notice of Allowance under 37 C.F.R. § 1.311. Please charge Deposit Account 50-3581 (TPO.004.C2) the amount of $180.00 under 37 C.F.R. § 1.17(p) for consideration of the items.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. § 1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.232.6907

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti
Reg. No. 57,125

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	2/18/2011	Office Action dated 2/18/2011 from U.S. Serial No. 11/338,996, 18 pages.
	5/17/2011	Office Action Response dated 5/17/2011 from U.S. Serial No. 11/338,996, 9 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

Electronic Patent Application Fee Transmittal

Application Number:	11338934
Filing Date:	25-Jan-2006
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Filer:	Erin M. Nichols
Attorney Docket Number:	TPO.004.C2

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
		Total in USD ($)		**180**

Electronic Acknowledgement Receipt

EFS ID:	10216166
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	10888
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	02-JUN-2011
Filing Date:	25-JAN-2006
Time Stamp:	13:21:50
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$180
RAM confirmation Number	13215
Deposit Account	503581
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		OAR-TPO004C2-OA04032011.pdf	83410 099487a97bfd6262ec23078255dae2f97c503adb	yes	9

Multipart Description/PDF files in .zip description		
Document Description	**Start**	**End**
Amendment/Req. Reconsideration-After Non-Final Reject	1	1
Claims	2	6
Applicant Arguments/Remarks Made in an Amendment	7	9

Warnings:

Information:

2	Transmittal Letter	IDS-14-TPO004C2.pdf	56107 4edd1bd0b511ff1d6b1ffab066d6d14388288b44	no	2

Warnings:

Information:

3	Information Disclosure Statement (IDS) Form (SB08)	1449-14-TPO004C2.pdf	52216 19716cc11fd61404e57b52456912382b253711de	no	1

Warnings:

Information:

This is not an USPTO supplied IDS fillable form

4	Non Patent Literature	OA-180202011.pdf	610274 23af7b687ed78aed21d21653afa8dda4fd02ba4d	no	18

Warnings:

Information:

5	Non Patent Literature	OAR-TPO004C1-OA18022011.pdf	79696 47300f709550cb902f658bd15bd927096f7e4390	no	9

Warnings:

Information:

6	Fee Worksheet (SB06)	fee-info.pdf	30217 ecad26660c9a3db990345b382663d3099b9bdf94	no	2

Warnings:

Information:

Total Files Size (in bytes):	911920

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
11/338,934	01/25/2006	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT 06/02/2011	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 11	Minus	** 20	= 0	X $ =		OR	X $52=	0
Independent (37 CFR 1.16(h))	* 6	Minus	***6	= 0	X $ =		OR	X $220=	0
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/ERIC DAVIS/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	2002	Hannuksela, "Signaling of Enhanced GOPs", Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 4th Meeting, Austria, July 22-26, 2002.
	4/5/2011	File History for EP Application No. 03700321.7 as retrieved from the European Patent Office Electronic File System on 4/5/2011, 222 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,934	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. § 1.97(c)(1)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted after the mailing date of the first Office action on the merits but before the mailing date of a final Office action under 37 C.F.R. § 1.113 or a Notice of Allowance under 37 C.F.R. § 1.311.

The items listed on the enclosed Form 1449 were first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449. However, if fees are deemed necessary, authorization is given to charge Deposit Account 50-3581 (TPO.004.C2) for consideration of the information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), a copy of the non-U.S. patent/application documents listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that the reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. § 1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,
HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti
Reg. No. 57,125

Electronic Acknowledgement Receipt

EFS ID:	9810103
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	10888
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	05-APR-2011
Filing Date:	25-JAN-2006
Time Stamp:	19:03:48
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Filed (SB/08)	1449-13-TPO004C2.pdf	59869 5b8deecba38c717dbf226b58d991b55531b5bf7f	no	1

Warnings:	
Information:	

This is not an USPTO supplied IDS fillable form

2	NPL Documents	FH-EP037003217.pdf	11489599 eb3fbc40a44f1218f89d4b32e1668595b2498987	no	222
Warnings:					
Information:					
3	NPL Documents	Hannuksela-2.pdf	73745 d1af007096e859d2cd6d5cc1d112dbf8df536262	no	9
Warnings:					
Information:					
4	Transmittal Letter	IDS-13-TPO004C2.pdf	56596 dc8f81fd9b533eb00b878aef5680652538bf003b	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	11679809		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
11/338,934		2611	

OC000000046476016

Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 10888 on 11/30/2010

- **Correspondence Address**
- **Power of Attorney Address**

The address of record for Customer Number 10888 is:

10888
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

OK to enter
/SL/

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,934	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on February 2, 2011.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.116

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action mailed December 3, 2010, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.116.

Claims 1, 2, 4-6, 8-11, 13, and 14 remain pending in the application. Claims 1, 5, 9, 10, 13, and 14 have been amended. No claims have been added or canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE SPECIFICATION begin on page 2.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 3 of this paper.

REMARKS/ARGUMENTS begin on page 8.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,934	01/25/2006	Miska Hannuksela	TPO.004.C2	6610

10888 7590 03/04/2011

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

NOTIFICATION DATE	DELIVERY MODE
03/04/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

tdotter@hfiplaw.com
roswood@hfiplaw.com

Office Action Summary	Application No.	Applicant(s)	
	11/338,934	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	SIU M. LEE	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 February 2011*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-6,8-11,13 and 14* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-6,8-11,13 and 14* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *1/25/2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☒ Certified copies of the priority documents have been received in Application No. 10/306942.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Transitional After Final Practice

1. Since this application is eligible for the transitional procedure of 37 CFR 1.129(a), and the fee set forth in 37 CFR 1.17(r) has been timely paid, the finality of the previous Office action is hereby withdrawn pursuant to 37 CFR 1.129(a). Applicant's first submission after final filed on 12/3/2011 has been entered.

Response to Arguments

2. Applicant's arguments, see page 8-9, filed on 12/3/2011, with respect to the rejections of claims 5-8 and 13 under 35 U.S.C. § 101 and claims 5-8 and 13 under 35 U.S.C. § 112, 1st paragraph have been fully considered and are persuasive. Therefore, the rejection has been withdrawn.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1-2, 4-6, 8-11, 13-14 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Independent claims 1, 5, 9-10, 13-14 all recite in the preamble a limitation "at least one reference image frame is predictable from at least one previous image frame,

and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order". As the limitation states, the reference image frame is predictable from a previous image frame, therefore, it is essential for the previous image frame to be decoded before it can use to predict the reference image frame. It is unclear when the reference image frame is decoded earlier than the previous image frame, it would not be predict from the previous image frame since the reference image frame is earlier in decoding order. As stated in the abstract of the instant application, "In the decoding phase, the indication of at least one image frame is decoded from the video sequence, and the decoding of the video sequence is started from **said first image frame of the independent sequence**, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame"; that is the first I frame of the sequence is always being decode first. Therefore, it is unclear what "at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order" means.

Independent claims 9-10, 13-14 all recite in the preamble a limitation "resetting the identifier value for the indicated first image frame of the independent sequence"; as disclose in paragraph 0018 of the publication of the instant application (2006/0120451 A1), the resetting step of the identifier value for the indicated first picture of an independent sequence is performed in the encoder, however, claims 9-11, 13-14 are

directed to a decoding, it is unclear how the decoder reset the identifier value of the indicated first image frame of the independent sequence.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2).

(1) Regarding claim 1:

(claim 1 is rejected without the consideration of the limitation in the preamble because of the 112 rejection as explained in above)

Apostolopoulos discloses an encoder comprising:

encoding into the video sequence an indication of at least one image frame, which is the first image frame (encoded I1 frame as shown in figure 4), in decoding order, of the independent sequence (decoding of the P frames are depending on the I1 frame), said at least one reference image frame being included in the sequence (all P frames are predict from I1 frame, the examiner interprets the claimed at least one reference image frame as one of the P frames as shown in figure 4);

encoding identifier values for the image frames according to a numbering scheme (the P frames are decoded sequentially from I1, column 9, lines 60-65; the numbering scheme of odd 1,3,5….15 and even 2,4,6…).

Apostolopoulos fails to explicitly disclose resetting the identifier value for the indicated first image frame of the independent sequence.

However, each sequence that is encoded is independent and it would have been obvious to one of ordinary skill in the art that it is not necessary to incrementing the identifier of the first image frame because it prevent using extra bit to represent a higher number of identifier and reduce the amount of data to be transmitted to the decoder.

(2) Regarding claim 10:

(claim 10 is rejected without the consideration of the limitation in the preamble because of the 112 rejection as explained in above, the examiner interprets the resetting of the identifier values are reset in the encoder)

Apostolopoulos discloses a method for decoding comprising:

decoding from the video sequence an indication of at least one image frame, which is the first image frame (the decoder decode the even frames and odd frames sequence, column 8, lines 40-49 as disclose in figure 4, the first frame is identified as an I1 frame for prediction of the following frames, column 9, lines 60-65), in decoding order, of an independent sequence, said sequence including the at least one reference image frame (all P frames are predict from I1 frame, the examiner interprets the claimed at least one reference image frame as one of the P frames as shown in figure 4);

starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame (all the other P frames are depend on the I1 frame, therefore, no prediction is need from any image frame prior to I1, column 9, lines 60-65);

decoding identifier values for image frames according to a numbering scheme (the P frames are decoded sequentially from I1, column 9, lines 60-65).

Apostolopoulos fails to explicitly disclose resetting the identifier value for the indicated first image frame of the independent sequence.

However, each sequence that is encoded is independent and it would have been obvious to one of ordinary skill in the art that it is not necessary to incrementing the identifier of the first image frame because it prevent using extra bit to represent a higher number of identifier and reduce the amount of data to be transmitted to the decoder.

7. Claims 2 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) as applied to claims 1 and 10 above, and further in view of Yagasaki et al. (US 5,786,858).

Apostolopoulos discloses all subject matter as discussed in claim 10 except wherein the indication is a separate flag included in the header of a slice.

However, Yagasaki discloses putting a flag for indicating an intra-frame coded block in the slice header (column 10, line 65 – column 11, line 2).

It is desirable to have the indication is a separate flag included in the header of a slice because it prevents a decoding process of the entire bit stream for extraction of the flag (column 11, lines 1-2). Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the teaching of Yagasaki in the decoder of Apostolopoulos to simplify the extraction process of the identifier.

8. Claims 5, 9, and 13-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) as applied to claim 10 above, and further in view of Langberg et al. (US 5,852,630).

(1) Regarding claims 5 and 9:

Apostolopoulos discloses an encoder comprising:

encoding into the video sequence an indication of at least one image frame, which is the first image frame (encoded I1 frame as shown in figure 4), in decoding order, of the independent sequence (decoding of the P frames are depending on the I1 frame), said at least one reference image frame being included in the sequence (all P frames are predict from I1 frame, the examiner interprets the claimed at least one reference image frame as one of the P frames as shown in figure 4);

encoding identifier values for the image frames according to a numbering scheme (the P frames are decoded sequentially from I1, column 9, lines 60-65; the numbering scheme of odd 1,3,5….15 and even 2,4,6…).

Apostolopoulos fails to explicitly disclose (a) resetting the identifier value for the indicated first image frame of the independent sequence; and (b) a computer program

product stored on a non-transitory computer readable medium and executable in a data processing device for implementing the method.

With respect to (a), each sequence that is encoded is independent and it would have been obvious to one of ordinary skill in the art that it is not necessary to incrementing the identifier of the first image frame because it prevent using extra bit to represent a higher number of identifier and reduce the amount of data to be transmitted to the decoder.

With respect to (b), Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system (processor) or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Apostolopoulos would have been implemented in a software. The implemented software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg et al. for implementing the decoding method of Apostolopoulos in order to reduce cost and improve the adaptability and flexibility of the encoder.

(2) Regarding claim 13 and 14:

Apostolopoulos discloses a method for decoding comprising:

decoding from the video sequence an indication of at least one image frame, which is the first image frame (the decoder decode the even frames and odd frames sequence, column 8, lines 40-49 as disclose in figure 4, the first frame is identified as an I1 frame for prediction of the following frames, column 9, lines 60-65), in decoding order, of an independent sequence, said sequence including the at least one reference image frame (all P frames are predict from I1 frame, the examiner interprets the claimed at least one reference image frame as one of the P frames as shown in figure 4);

starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame (all the other P frames are depend on the I1 frame, therefore, no prediction is need from any image frame prior to I1, column 9, lines 60-65);

decoding identifier values for image frames according to a numbering scheme (the P frames are decoded sequentially from I1, column 9, lines 60-65).

Apostolopoulos fails to explicitly disclose (a) resetting the identifier value for the indicated first image frame of the independent sequence; and (b) the decoder comprising at least one processor and at leas tone memory including computer program code to perform the method.

With respect to (a), each sequence that is encoded is independent and it would have been obvious to one of ordinary skill in the art that it is not necessary to incrementing the identifier of the first image frame because it prevent using extra bit to

represent a higher number of identifier and reduce the amount of data to be transmitted to the decoder.

with respect to (b), Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system (processor) or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Apostolopoulos would have been implemented in a software. The implemented software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg et al. for implementing the decoding method of Apostolopoulos in order to reduce cost and improve the adaptability and flexibility of the decoder.

9. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) as applied to claim 10 above, and further in view of Yagasaki et al. (US 5,786,858).

Apostolopoulos discloses all subject matter as discussed in claim 10 except wherein the indication is a separate flag included in the header of a slice.

However, Yagasaki discloses putting a flag for indicating an intra-frame coded block in the slice header (column 10, line 65 – column 11, line 2).

It is desirable to have the indication is a separate flag included in the header of a slice because it prevents a decoding process of the entire bit stream for extraction of the flag (column 11, lines 1-2). Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the teaching of Yagasaki in the decoder of Apostolopoulos to simplify the extraction process of the identifier.

10. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) in view of Langberg et al. (US 5,852,630) as applied to claim 5 above, and further in view of Yagasaki et al. (US 5,786,858).

Apostolopoulos and Langberg discloses all subject matter as discussed in claim 5 except wherein the indication is a separate flag included in the header of a slice.

However, Yagasaki discloses putting a flag for indicating an intra-frame coded block in the slice header (column 10, line 65 – column 11, line 2).

It is desirable to have the indication is a separate flag included in the header of a slice because it prevents a decoding process of the entire bit stream for extraction of the flag (column 11, lines 1-2). Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the teaching of Yagasaki in the decoder of Apostolopoulos and Langberg to simplify the extraction process of the identifier.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Hasebe et al. (US 6,483,875 B1) discloses a picture signal processing apparatus.

Shinoda (US 5,680,322) discloses a method and apparatus for dynamic image data transmission.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU M. LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 7:30-4:00 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Examiner, Art Unit 2611
2/28/2011

Notice of References Cited	Application/Control No. 11/338,934	Applicant(s)/Patent Under Reexamination HANNUKSELA, MISKA	
	Examiner SIU M. LEE	Art Unit 2611	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-5,680,322	10-1997	Shinoda, Mayumi	714/18
*	B	US-6,483,875	11-2002	Hasebe et al.	375/240.15
*	C	US-5,786,858	07-1998	Yagasaki et al.	375/240.15
*	D	US-7,103,669	09-2006	Apostolopoulos, John G.	709/231
*	E	US-5,852,630	12-1998	Langberg et al.	375/219
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

To: tdotter@hfiplaw.com,roswood@hfiplaw.com,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 10888

Mar 04, 2011 06:00:56 AM

Dear PAIR Customer:

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 10888 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
11338934	CTNF	03/04/2011	TPO.004.C2
	892	03/04/2011	TPO.004.C2

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,934	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on February 2, 2011.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.116

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action mailed December 3, 2010, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.116.

Claims 1, 2, 4-6, 8-11, 13, and 14 remain pending in the application. Claims 1, 5, 9, 10, 13, and 14 have been amended. No claims have been added or canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE SPECIFICATION begin on page 2.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 3 of this paper.

REMARKS/ARGUMENTS begin on page 8.

AMENDMENTS TO THE SPECIFICATION

Please amend the paragraph at page 33, lines 6-16, of the original specification as indicated below.

[0097] It is to be noted that according to the invention the functional elements of the above described streaming system and its elements, such as the streaming server, video encoder, video decoder and terminal are preferably implemented by ~~means of software, by~~ hardware solutions[[,]] or as a combination of ~~the two~~hardware and software. The coding and decoding methods of the invention are particularly suitable for implementation as computer software comprising computer-readable commands for executing the process steps of the invention. A preferred way of implementing the encoder and the decoder is to store them in a storage means as a program code that can be executed by a computer-like device, for example a personal computer (PC) or a mobile station, to provide coding/decoding functionalities on the device in question.

AMENDMENTS TO THE CLAIMS

Applicant respectfully requests that the amendments presented below be entered and that this listing of claims replace all prior versions of claims in the application.

1. (Currently amended) A method for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order, the method comprising:

encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence;

encoding identifier values for the image frames according to a numbering scheme; and

resetting the identifier value for the indicated first image frame of [[an]]the independent sequence.

2. (Original) A method according to claim 1, further comprising:

encoding the indication into the video sequence as a separate flag included in the header of a slice.

3. (Canceled)

4. (Original) A method according to claim 1, further comprising:

encoding into the video sequence an identifier value for the independent sequence.

5. (Currently amended) A video encoder ~~for encoding~~comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video

encoder to encode a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference frame is earlier than the at least one previous image frame in decoding order~~, the video encoder being configured to~~by causing the video encoder to at least:

encode into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence;

encode identifier values for the image frames according to a numbering scheme; and

reset the identifier value for the indicated first image frame of [[an]]the independent sequence.

6. (Previously presented) A video encoder according to claim 5, wherein the video encoder is configured to

encode the indication into the video sequence as a separate flag included in the header of a slice.

7. (Canceled)

8. (Previously presented) A video encoder according to claim 5, wherein the video encoder is configured to

encode into the video sequence an identifier value for the independent sequence.

9. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame and wherein the at least one reference frame is earlier than the at least one previous image frame in decoding order, the computer program product comprising:

a computer program code for encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence;

a computer code for encoding identifier values for the image frames according to a numbering scheme; and

a computer code for resetting the identifier value for the indicated first image frame of [[an]]the independent sequence.

10. (Currently amended) A method for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame and wherein the at least one reference frame is earlier than the at least one previous image frame in decoding order, the method comprising:

decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame;

starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;

decoding identifier values for image frames according to a numbering scheme; and

resetting the identifier value for the indicated first image frame of [[an]]the independent sequence.

11. (Original) A method according to claim 10, wherein the indication is a separate flag included in the header of a slice.

12. (Canceled)

13. (Currently amended) A video decoder ~~for decoding~~comprising at least one processor and at least one memory including computer program code, the at least one memory and the

computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order~~, the video decoder being configured to~~by causing the video decoder to at least:

decode from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame;

start the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame; and

decode identifier values for image frames according to a numbering scheme; and

reset the identifier value for the indicated first image frame of [[an]]the independent sequence.

14. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order, the computer program product comprising:

a computer program code for decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame;

a computer program code for starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;

a computer program code for decoding identifier values for image frames according to a numbering scheme; and

a computer program code for resetting the identifier value for the indicated first image frame of [[an]]the independent sequence.

15-16. (Canceled)

REMARKS/ARGUMENTS

The rejections presented in the Office action dated December 3, 2010 (hereinafter Office Action) have been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

With respect to the objection to claims 1, 2, 4-6, 8, 10, and 11, claims 1, 5, and 9 have been amended to provide explicit antecedent basis for "the image frames" in accordance with the Examiner's suggestions. Also, claims 1, 5, 9, and 10 have been amended to provide explicit antecedent basis for the claimed independent sequence. Although not identified in the statement of objection, claims 13 and 14 have also been amended to provide explicit antecedent basis for the claimed independent sequence. These changes are believed to overcome the objection; therefore, Applicant requests that the objection be removed.

In view of the above, at least claims 1, 2, 4, 9-11, and 14 are now believed to be in condition for allowance, and Applicant accordingly requests that these claims be allowed consistent with the assertions at page seven *et seq*.

With respect to the § 101 rejections of claims 5, 6, 8, and 13, Applicant respectfully traverses. As acknowledged at page ten of the Office action, claims 5 and 13 are directed to components. While the specification also refers to embodiments involving computer readable media storing program code (*e.g.*, claims 9 and 14), the claimed encoder and decoder are devices constituting statutory subject matter. The terms "encoder" and "decoder" are well-known in the art to refer to a device or circuit. Notably, the specification illustrates structural components of the encoder (*e.g.*, compressor 108 and multiplexer 110) and decoder (*e.g.*, demultiplexer 116 and decompressor 118). Paragraph [0095] teaches examples of a decoder including a low bit rate video decoder as described in the ITU-T recommendation H.263 or H.26L. Moreover, paragraph [0096] describes positioning of the encoder and decoder that would not be possible if they were only software. However, paragraph [0097] has also been amended to remove an interpretation of the claimed encoder and decoder as being software only. Since the claimed encoder and

decoder are directed to statutory subject matter, Applicant requests that the rejection be withdrawn.

With respect to the § 112, first paragraph, rejection of claims 5, 6, 8, and 13, independent claims 5 and 13 have been amended to recite that the respective encoder and decoder comprise at least one processor and at least one memory. Support for these changes may be found in the original specification, for example, at paragraphs [0088], [0089], [0093], [0096], and [0097]; therefore, these changes do not introduce new matter. Since the device claims each recite more than a single component, the rejection is believed to be overcome. Applicant accordingly requests that the rejection be withdrawn.

In view of the above, Applicant believes each of the pending claims is in condition for allowance and requests that the application be allowed.

Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C2) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the Examiner is invited to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.232.6907

Date: February 2, 2011

By: /Erin Nichols Matkaiti/

Erin Nichols Matkaiti
Reg. No. 57,125

Electronic Acknowledgement Receipt

EFS ID:	9364608
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	10888
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	02-FEB-2011
Filing Date:	25-JAN-2006
Time Stamp:	18:59:12
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		OAR-TPO004C2-OA03122010.pdf	87566 573dcb95a8e974414fe4422090151ea0daabee3b	yes	9

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment After Final	1	1
	Specification	2	2
	Claims	3	7
	Applicant Arguments/Remarks Made in an Amendment	8	9
Warnings:			
Information:			
	Total Files Size (in bytes):	87566	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
11/338,934	01/25/2006	

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY ☐ RATE ($)	SMALL ENTITY ☐ FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
FOR	NUMBER FILED	NUMBER EXTRA					
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT 02/02/2011

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 11	Minus	** 20	= 0	X $ =		OR	X $52=	0
Independent (37 CFR 1.16(h))	* 6	Minus	***6	= 0	X $ =		OR	X $220=	0
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/TRACIE ROBERTSON/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,934	01/25/2006	Miska Hannuksela	TPO.004.C2	6610

10888 7590 12/03/2010

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

MAIL DATE	DELIVERY MODE
12/03/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/338,934	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	SIU M. LEE	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 September 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-6,8-11,13 and 14* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *5,6,8 and 13* is/are rejected.

7)☒ Claim(s) *1,2,4,9-11 and 14* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 January 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☒ Certified copies of the priority documents have been received in Application No. 10/306,942.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *See Continuation Sheet*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :8/10/2010, 9/30/2010, 10/14/2010.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments with respect to the 35 U.S.C. 101 rejection of claims 5-8 and 13, filed 9/17/2010 have been fully considered but they are not persuasive.

Applicant's argument:

The claimed encoder and decoder are well known in the art to refer to a device or circuit; the specification illustrates structural component of the encoder (e.g. compressor 108 and multiplexer 110) and decoder (e.g. demultiplexer 116 and decompressor 118). Paragraph 0095 teaches example of a decoder including a low bit rate video decoder as described in the ITU-T recommendation H.263 or H.26L. Moreover, paragraph 0096 describes positioning of the encoder and decoder that would not be possible if they were "pure software".

Examiner's response:

The following recites paragraph 0104 of the publication of the instant application (US 2006/0120451 A1):

> [0104] It is to be noted that according to the invention the functional elements of the above described streaming system and its elements, such as the streaming server, video encoder, video decoder and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two. The coding and decoding methods of the invention are particularly suitable for implementation as computer software comprising computer-readable commands for executing the process steps of the invention. A preferred way of implementing the encoder and the decoder is to store them in a storage means as a program code that can be executed by a computer-like device, for example a personal computer (PC) or a mobile station, to provide coding/decoding functionalities on the device in question.

Paragraph 0104 discloses that the video encoder, video decoder and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two. Since claim 5 is non-structural per se and is only limited by the functions, therefore, a reasonable broadest interpretation in light of the specification leads to the conclusion that the claimed video encoder covers a hardware embodiment (statutory) as well as a software embodiment (non-statutory), which does not fall within the definition of a process, machine, manufacture, or composition of matter.

With regard to the argument that the specification illustrates structural component of the encoder (e.g. compressor 108 and multiplexer 110), the components are represented by block in figure 1 and only define by the function as disclose in the specification, therefore, it does not exclude a software implementation of these components.

With respect to the argument that Paragraph 0095 teaches example of a decoder including a low bit rate video decoder as described in the ITU-T recommendation H.263

or H.26L. Moreover, paragraph 0096 describes positioning of the encoder and decoder that would not be possible if they were "pure software"; however, it is discloses in the above quoted paragraph (paragraph 0104 of the publication of the instant application (US 2006/0120451 A1)) that encoder and decoder can be implemented as software, as the claim is not structural per se, the examiner interprets the claim covers a software embodiment as well, therefore the rejection is maintained. If the applicant considered that the encoder and decoder can not be “pure software’, the examiner suggests amending paragraph 0104 of the publication of the instant application to remove the software embodiment.

2. Applicant's arguments with respect to the 35 U.S.C. 112, 1st paragraph rejection of claims 5-6, 8, and 13 filed 9/17/2010 have been fully considered but they are not persuasive.

Applicant argument:

Applicant respectfully traverses the § 112, first paragraph, rejection of claims 5, 6, 8, and 13 because the claims are not "single means" claims. First, none of these claims uses a "means recitation" or "means plus function" language. While a claim dependent upon a recited property could result in a fact situation similar to that discussed in the cited Hyatt decision when the claim covers every conceivable structure for achieving the stated property, such is not the case for claims 5, 6, 8, and 13. In contrast the claims are limited to the structures of an encoder (claims 5, 6, and 8) and of a decoder (claim 13), which are well-accepted and understood terms in the art, for

achieving the stated properties of the respective claims. Therefore, the claims are not unduly broad and are enabled by the specification. Applicant accordingly requests that the rejection be withdrawn.

Examiner's response:

The examiner would like to point out that claim 5-6, 8, and 13 are not rejection as a single means claim rejection. Claims 5-6, 8, and 13 are rejected based on the claim only directed to a single structure or component (decoder) and the decoder is defined by the recited properties (functions of the decoder), than a fact situation similar to that discussed in the cited Hyatt decision when the claim covers every conceivable structure for achieving the stated property for achieving the stated property while the specification discloses at most only known to the invention. Therefore, the 35 U.S.C. 112, 1st paragraph rejections of claims 5-6, 8, and 13 are maintained.

3. Applicant's arguments, see page 7-9, filed 9/17/2010, with respect to 35 U.S.C. 102(e) and 103(a) rejections of claims 1-16 have been fully considered and are persuasive. The 35 U.S.C. 102(e) and 103(a) rejections of claims 1-16 has been withdrawn.

With regard to the double patenting rejection of claims 1-2 and 4 are withdrawn because of the combination of claim 3 into claim 1.

Claim Objections

4. Claims 1-2, 4-6, 8, 10-11, are objected to because of the following informalities:

Claim 1, line 9 recites "encoding identifier values for image frames"; since "image frames" was introduced in line 2, the examiner suggests changing to "encoding identifier values for **the** image frames".

Claim 1, line 10-11 recite "resetting the identifier value for the indicated first image frame of an independent sequence"; since "independent sequence" was introduced in line 2, it is unclear whether it is the same independent sequence or another independent sequence.

Claim 5, line 9 recites "encoding identifier values for image frames"; since "image frames" was introduced in line 2, the examiner suggests changing to "encoding identifier values for **the** image frames".

Claim 5, line 10-11 recite "resetting the identifier value for the indicated first image frame of an independent sequence"; since "independent sequence" was introduced in line 2, it is unclear whether it is the same independent sequence or another independent sequence.

Claim 9, line 10 recites "encoding identifier values for image frames"; since "image frames" was introduced in line 3, the examiner suggests changing to "encoding identifier values for **the** image frames".

Claim 9, line 12-13 recite "resetting the identifier value for the indicated first image frame of an independent sequence"; since "independent sequence" was introduced in line 2, it is unclear whether it is the same independent sequence or another independent sequence.

Claim 10, lines 12-13 recite "resetting the identifier value for the indicated first image frame of an independent sequence"; since "independent sequence" was introduced in line 6, it is unclear whether it is the same independent sequence or another independent sequence.

Claim 13, lines 13-14 recite "resetting the identifier value for the indicated first image frame of an independent sequence"; since "independent sequence" was introduced in line 7, it is unclear whether it is the same independent sequence or another independent sequence.

Claim 14, lines 15-16 recite "resetting the identifier value for the indicated first image frame of an independent sequence"; since "independent sequence" was introduced in lines 8-9, it is unclear whether it is the same independent sequence or another independent sequence.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 5-8 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter.

The following recites paragraph 0104 of the publication of the instant application (US 2006/0120451 A1):

> [0104] It is to be noted that according to the invention the functional elements of the above described streaming system and its elements, such as the streaming server, video encoder, video decoder and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two. The coding and decoding methods of the invention are particularly suitable for implementation as computer software comprising computer-readable commands for executing the process steps of the invention. A preferred way of implementing the encoder and the decoder is to store them in a storage means as a program code that can be executed by a computer-like device, for example a personal computer (PC) or a mobile station, to provide coding/decoding functionalities on the device in question.

Paragraph 0104 discloses that the video encoder, video decoder and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two. Since the encoder of claim 5 is non-structural per se and is only limited by the functions, therefore, a reasonable broadest interpretation in light of the specification leads to the conclusion that the claimed video encoder covers a hardware embodiment (statutory) as well as a software embodiment (non-statutory), which does not fall within the definition of a process, machine, manufacture, or composition of matter.

Claim 13 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter.

The following recites paragraph 0104 of the publication of the instant application (US 2006/0120451 A1):

> [0104] It is to be noted that according to the invention the functional elements of the above described streaming system and its elements, such as the streaming server, video encoder, video decoder and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two. The coding and decoding methods of the invention are particularly suitable for implementation as computer software comprising computer-readable commands for executing the process steps of the invention. A preferred way of implementing the encoder and the decoder is to store them in a storage means as a program code that can be executed by a computer-like device, for example a personal computer (PC) or a mobile station, to provide coding/decoding functionalities on the device in question.

Paragraph 0104 discloses that the video encoder, video decoder and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two. Since the decoder of claim 13 is non-structural per se and is only limited by the functions, therefore, a reasonable broadest interpretation in light of the specification leads to the conclusion that the claimed video decoder covers a hardware embodiment (statutory) as well as a software embodiment (non-statutory), which does not fall within the definition of a process, machine, manufacture, or composition of matter.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 5-8 are rejected under 112, 1st paragraph as being a single component (video encoder) claim, i.e., where a single component (video encoder) recitation does not appear in combination with another recited component, is subject to an undue breadth rejection. The single component which covered every conceivable ways for achieving the stated purpose was held nonenabling for the scope of claim because the specification disclosed at most only those known to the inventor. See MPEP §2164.08(a).

Claim 13 is rejected under 112, 1st paragraph as being a single component (video decoder) claim, i.e., where a single component (video decoder) recitation does not appear in combination with another recited component, is subject to an undue breadth rejection. The single component which covered every conceivable ways for achieving the stated purpose was held nonenabling for the scope of claim because the specification disclosed at most only those known to the inventor. See MPEP §2164.08(a).

Allowable Subject Matter

7. Claims 1-2, 4, 9-11, and 14 are objected to because of minor informality, but would be allowable if rewritten to overcome the objection.

8. The following is a statement of reasons for the indication of allowable subject matter:

The present invention describes a method for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference

image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame is earlier than the at least one previous image frame in decoding order, the method comprising encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence; encoding identifier values for image frames according to a numbering scheme; and resetting the identifier value for the indicated first image frame of an independent sequence. the closest prior art, Kim in view of Ueda and Kawashima together disclose a similar method but fails to disclose encoding identifier values for image frames according to a numbering scheme; and resetting the identifier value for the indicated first image frame of an independent sequence.

The present invention describes a method for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame and wherein the at least one reference frame is earlier than the at least one previous image frame in decoding order, the method comprising decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame; starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame; decoding identifier values for image frames according to a numbering scheme; and resetting the identifier value for the indicated first image frame of an independent

sequence. The closest prior art, Kim in view of Ueda and Kawashima together disclose a similar method but fails to disclose decoding identifier values for image frames according to a numbering scheme; and resetting the identifier value for the indicated first image frame of an independent sequence.

Conclusion

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU M. LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 7:30-4:00 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Examiner, Art Unit 2611
12/1/2010

/CHIEH M FAN/

Supervisory Patent Examiner, Art Unit 2611

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/SL/	5852630	12/22/1998	Langberg			
/SL/	6072831	06/06/2000	Chen			
/SL/	6363208	03/26/2002	Nitta et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	2137197C1	9/10/1999	Russia				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	1997	Illgner et al., "Spatially Scalable Video Compression Employing Resolution Pyramids", IEEE Journal on Selected Areas in Communications, Vol. 15, No. 9, December 1, 2007, pg. 1688-1703.
/SL/	2000	Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Systems, ISO/IEC 13818-1, Second edition, December 1, 2000, 174 pages.
/SL/	5/5/2010	Notice of Allowance dated 5/5/2010 from Russian Application No. 2006110321/09, 11 pages.
/SL/	6/17/2010	Office Action dated 6/17/2010 from U.S. Serial No. 11/338,996, 30 pages.
/SL/	6/23/2010	Office Action dated 6/23/2010 from U.S. Serial No. 10/306,942, 6 pages.
/SL/	5/25/2010	Office Action dated 5/25/2010 from U.S. Serial No. 10/306,942, 6 pages.
/SL/	4/09/2009	Office Action dated 4/9/2009 from U.S. Serial No. 10/306,942, 4 pages.
/SL/	5/8/2006	Office Action Response dated 5/8/2006 from U.S. Serial No. 10/306,942, 11 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 12/01/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	JP11275576	10/08/1999	JP			Abstract only	
/SL/	JP4348690	12/03/1992	JP			Abstract only	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	08/17/2010	Copy of translated Office Action dated 08/17/2010 from parallel Japanese Application No. 2006-120297, 4 pages.
/SL/	08/24/2010	Copy of translated Office Action dated 08/24/2010 from parallel Japanese Application No. 2006-120296, 5 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 12/01/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	9/15/2010	Office Action Response dated 9/15/2010 from U.S. Serial No. 10/306,942, 12 pages.
/SL/	9/10/2010	Office Action Response dated 9/10/2010 from U.S. Serial No. 11/338,996, 10 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 12/01/2010

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	JP11275576	10/08/1999	JP			Abstract only	
	JP4348690	12/03/1992	JP			Abstract only	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	08/17/2010	Copy of translated Office Action dated 08/17/2010 from parallel Japanese Application No. 2006-120297, 4 pages.
	08/24/2010	Copy of translated Office Action dated 08/24/2010 from parallel Japanese Application No. 2006-120296, 5 pages.

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,934	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. §1.97(c)(1) and 37 C.F.R. §1.97 (e)(1)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted after the mailing date of the first Office Action on the merits but before the mailing date of a final Office Action under 37 C.F.R. §1.113 or a Notice of Allowance under 37 C.F.R. §1.311.

The items listed on the enclosed Form 1449 were first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449. However, if fees are deemed necessary, authorization is given to charge Deposit Account 50-3581 (TPO.004.C2) for consideration of the information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. § 1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Steven R. Funk/
Steven R. Funk
Reg. No. 37,830

Electronic Acknowledgement Receipt

EFS ID:	8624884
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Steven R. Funk
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	14-OCT-2010
Filing Date:	25-JAN-2006
Time Stamp:	14:33:35
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Filed (SB/08)	Form1449-12TPO004C2.pdf	71878 07c1f412c9687312dbfb7b4a384b8258d601a372	no	1

Warnings:	
Information:	

This is not an USPTO supplied IDS fillable form

2	Foreign Reference	JP11275576pre.pdf	236628 f872f28d2862519b5f1d6ad930ba7863892ec8e3	no	35
Warnings:					
Information:					
3	Foreign Reference	JP4348690pre.pdf	188895 cabd1429d49abfe3242ac88d82326510350983ea	no	6
Warnings:					
Information:					
4	NPL Documents	JPOA20100817.pdf	74168 f318ac7a8fbcd4fe55a3d0fd18432cfe25dd0936	no	4
Warnings:					
Information:					
5	NPL Documents	JPOA20100824.pdf	94278 a850e4ac8e953ae55e3060329b123b4f4e6e0fda	no	5
Warnings:					
Information:					
6	Transmittal Letter	IDS-12-TPO004C2.pdf	57754 e6ab8a976ba1a5f7718b6cdd231ab7eeab1a13a4	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	723601		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	9/15/2010	Office Action Response dated 9/15/2010 from U.S. Serial No. 10/306,942, 12 pages.
	9/10/2010	Office Action Response dated 9/10/2010 from U.S. Serial No. 11/338,996, 10 pages.

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,934	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. § 1.97(c)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant respectfully submits the items of information on the enclosed Form 1449 for the attention of the Examiner in the above-identified application.

This statement should be considered because it is submitted after the mailing date of the first Office action on the merits but before the mailing date of a final Office action under 37 C.F.R. § 1.113 or a Notice of Allowance under 37 C.F.R. § 1.311. Please charge Deposit Account 50-3581 (TPO.004.C2) the amount of $180.00 under 37 C.F.R. § 1.17(p) for consideration of the items.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. § 1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti
Reg. No. 57,125

Electronic Patent Application Fee Transmittal

Application Number:	11338934
Filing Date:	25-Jan-2006
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Filer:	Erin M. Nichols
Attorney Docket Number:	TPO.004.C2

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
		Total in USD ($)		**180**

Electronic Acknowledgement Receipt

EFS ID:	8531082
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	30-SEP-2010
Filing Date:	25-JAN-2006
Time Stamp:	15:19:31
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$180
RAM confirmation Number	4605
Deposit Account	503581
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Filed (SB/08)	1449-11-TPO004C2.pdf	51214 1fa897d699f527cf94be8e8ae5e3eb24ec21e7b5	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
2	NPL Documents	OAR-TAPA004PA-OA23062010.pdf	97281 8e13d6c942b64fe9bc59ac82106b7e9c0b09ce2a	no	12
Warnings:					
Information:					
3	NPL Documents	OAR-TPO004C1-OA17062010.pdf	89215 26aba7cc752c3fc6ba6ddd2ef825b135aa56074e	no	10
Warnings:					
Information:					
4	Transmittal Letter	IDS-11-TPO004C2.pdf	55429 ca0531d1d8e8565dfc486c03a855f3ba7796a8f1	no	2
Warnings:					
Information:					
5	Fee Worksheet (PTO-875)	fee-info.pdf	30218 c426fea1268d247a4eb953a81801d64fed4fdb26	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	323357		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,934	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on September 17, 2010.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action mailed May 25, 2010, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.111.

Claims 1, 2, 4-6, 8-11, 13, and 14 remain pending in the application. Claims 1, 5, 6, 8-10, 13, and 14 have been amended. No claims have been added, and claims 3, 7, 12, 15, and 16 have been canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 6.

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions of claims in the application.

1. (Currently amended) A method for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame ~~that~~ is earlier than the at least one previous ~~reference~~ image frame in decoding order, the method comprising:

encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence;

encoding identifier values for image frames according to a numbering scheme; and

resetting the identifier value for the indicated first image frame of an independent sequence.

2. (Original) A method according to claim 1, further comprising:

encoding the indication into the video sequence as a separate flag included in the header of a slice.

3. (Canceled)

4. (Original) A method according to claim 1, further comprising:

encoding into the video sequence an identifier value for the independent sequence.

5. (Currently amended) A video encoder for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference frame ~~that~~ is earlier than the at least one previous ~~reference~~ image frame in decoding order, the video encoder being ~~arranged~~configured to:

encode into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence;

encode identifier values for image frames according to a numbering scheme; and

reset the identifier value for the indicated first image frame of an independent sequence.

6. (Currently amended) A video encoder according to claim 5, wherein the video encoder is ~~arranged~~configured to

encode the indication into the video sequence as a separate flag included in the header of a slice.

7. (Canceled)

8. (Currently amended) A video encoder according to claim 5, wherein the video encoder is ~~arranged~~configured to

encode into the video sequence an identifier value for the independent sequence.

9. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame [[that]]and wherein the at least one reference frame is earlier than the at least one previous ~~reference~~ image frame in decoding order, the computer program product comprising:

a computer program code for encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence;

a computer code for encoding identifier values for image frames according to a numbering scheme; and

a computer code for resetting the identifier value for the indicated first image frame of an independent sequence.

10. (Currently amended) A method for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame [[that]]and wherein the at least one reference frame is earlier than the at least one previous ~~reference~~ image frame in decoding order, the method comprising:

decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame; [[and]]

starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;

decoding identifier values for image frames according to a numbering scheme; and

resetting the identifier value for the indicated first image frame of an independent sequence.

11. (Original) A method according to claim 10, wherein the indication is a separate flag included in the header of a slice.

12. (Canceled)

13. (Currently amended) A video decoder for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame ~~that~~ is earlier than the at least one previous ~~reference~~ image frame in decoding order, the video decoder being ~~arranged~~configured to:

decode from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame; [[and]]

start the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame; and

decode identifier values for image frames according to a numbering scheme; and

reset the identifier value for the indicated first image frame of an independent sequence.

14. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame, and wherein the at least one reference image frame ~~that~~ is earlier than the at least one previous ~~reference~~ image frame in decoding order, the computer program product comprising:

a computer program code for decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame; [[and]]

a computer program code for starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;

a computer program code for decoding identifier values for image frames according to a numbering scheme; and

a computer program code for resetting the identifier value for the indicated first image frame of an independent sequence.

15-16. (Canceled)

REMARKS/ARGUMENTS

The rejections presented in the Office action dated May 25, 2010 (hereinafter Office Action), have been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

Claims 3, 7, 12, 15, and 16 have been canceled, without prejudice, rendering the respective rejections of these claims moot. Applicant accordingly requests that the rejections of each of these claims be withdrawn.

With respect to the objection to claims 1, 2, 4, 5, 6, 8, 10, and 11, the term "the previous reference image frame" in independent claims 1, 5, and 10 has been amended to read "the previous image frame" in order to provide antecedent basis for the term. Applicant accordingly requests that the objection be removed.

With respect to the objection to Fig. 1, Applicant respectfully traverses. In contrast to the assertion that the figure includes "blank boxes", each of the boxes of Fig. 1 is labeled with a reference numeral in accordance with MPEP § 608.02. Moreover, no requirement has been cited for the request to label the boxes with text. Since Fig. 1 complies with the drawing requirements, and in an effort to maintain consistency with the parent application, Applicant requests that the objection be removed.

With respect to the § 101 rejection of claims 5-8 and 13, Applicant respectfully traverses. The Office action incorrectly asserts that the claimed encoder and decoder are "pure software". While the specification refers to embodiments involving computer readable media storing program code (*e.g.*, claims 9 and 14), the claimed encoder and decoder are devices constituting statutory subject matter. The terms "encoder" and "decoder" are well-known in the art to refer to a device or circuit. Notably, the specification illustrates structural components of the encoder (*e.g.*, compressor 108 and multiplexer 110) and decoder (*e.g.*, demultiplexer 116 and decompressor 118). Paragraph [0095] teaches examples of a decoder including a low bit rate video decoder as described in the ITU-T recommendation H.263 or H.26L. Moreover, paragraph [0096] describes positioning of the encoder and decoder that would not be possible if they were "pure

software". Contrary to the assertion, the claimed encoder and decoder are devices constituting statutory subject matter. Applicant accordingly requests that the rejection be withdrawn.

With respect to the § 101 rejection of claims 9 and 14, the claims have been amended to indicate that the claimed computer readable media are non-transitory in accordance with the Examiner's suggestion. Applicant notes that non-transitory computer readable media comprise all computer readable media, with the sole exception being a transitory, propagating signal. Since an interpretation of the claims is now limited to statutory subject matter, Applicant submits that the rejection has been overcome and accordingly requests that the rejection be withdrawn.

Applicant respectfully traverses the § 112, first paragraph, rejection of claims 5, 6, 8, and 13 because the claims are not "single means" claims. First, none of these claims uses a "means recitation" or "means plus function" language. While a claim dependent upon a recited property could result in a fact situation similar to that discussed in the cited *Hyatt* decision when the claim covers every conceivable structure for achieving the stated property, such is not the case for claims 5, 6, 8, and 13. In contrast the claims are limited to the structures of an encoder (claims 5, 6, and 8) and of a decoder (claim 13), which are well-accepted and understood terms in the art, for achieving the stated properties of the respective claims. Therefore, the claims are not unduly broad and are enabled by the specification. Applicant accordingly requests that the rejection be withdrawn.

Applicant respectfully traverses the § 102(e) rejection of claim 1 in view of U.S. Publication No. 2003/0063806 by Kim *et al*. (hereinafter "Kim") because the Office action acknowledges that Kim fails to teach or suggest each of the claimed limitations. In the § 103(a) rejection of claim 1 at page fifteen, the Office action states that "Kim fails to explicitly disclose an indication of the first I frame". Since Kim admittedly fails to correspond to each of the limitations of at least claim 1, the § 102(e) rejection is improper. Applicant accordingly requests that the rejection be withdrawn.

In an effort to facilitate prosecution and without acquiescing to characterizations of the asserted art, Applicant's claimed subject matter, or to the applications of the asserted art

or combinations thereof to Applicant's claimed subject matter, the independent claims have been amended to include the respective limitations of dependent claims 3, 7, and 12; therefore, these changes do not introduce new matter. Since the Office action acknowledges that Kim and U.S. Patent No. 6,094,456 to Ueda fail to teach or suggest each of the limitations of dependent claims 3, 7, and 12, each of the rejections based on these references is overcome. Each of the claims is further believed to be patentable over the asserted references for the reasons set forth below.

In rejecting dependent claims 3, 7, and 12, the Office action relies upon the teachings of U.S. Patent No. 4,390,966 to Kawashima *et al.* (hereinafter "Kawashima") as teaching the claimed resetting of the identifier value for the indicated first image frame of an independent sequence; however, this reliance is misplaced. Kawashima merely teaches that a sequence controller system is capable of setting and resetting flags according to predetermined user instructions. Kawashima's flags fail to at least correspond to the claimed identifier value since Kawashima is silent with respect to the flags being associated with an image frame of a coded video sequence. Also, Kawashima's flags are toggle flags for use in a multi CPU environment where each controller can run sequentially stored code in a parallel environment. There is no correspondence to the claimed resetting of an encoded identifier value for a first image frame of a video sequence. Without correspondence to each of the claimed limitations, the § 103(a) rejections of the subject matter of dependent claims 3, 7, and 12 is improper. Applicant accordingly requests that the rejections be withdrawn.

Further, the asserted modification of the combination of Kim and Ueda with the teachings of Kawashima is improper since the asserted modification is illogical and the teachings of Kawashima are not analogous to the field of Applicant's endeavor. First, the Office action asserts that it would be desirable to reset the identifier value since the video sequence is independent because it would simplify the identifier value and reset the identifier value. No explanation or evidence has been provided to indicate that resetting Kawashima's flags would simplify a value. Moreover, it is illogical to assert that it is

desirable to reset an identifier value because it would reset the identifier value. Thus, the proposed modification to the combination of Kim and Ueda is illogical and unsupported.

Second, the teachings of Kawashima are nonanalogous art as they are directed to very different technology than the field of video coding. In contrast, Kawashima is directed to the field of multi-core processing. Since Kawashima is in a field different than Applicant's endeavor, to be considered reasonably pertinent, Kawashima would need to address a problem addressed by the application at issue (MPEP § 2141.01(a)). However, the Office action fails to assert or identify any such relationship. Kawashima, directed to very different technology and failing to address video coding, would not have logically commended itself to an inventor of the instant application. Thus, Kawashima is nonanalogous art and fails to support the § 103(a) rejections. Applicant respectfully maintains the request that the § 103(a) rejections based upon Kawashima be withdrawn.

It should also be noted that Applicant does not acquiesce to the Examiner's statements or conclusions concerning what would have been obvious to one of ordinary skill in the art, obvious design choices, common knowledge at the time of Applicant's invention, officially noticed facts, and the like. Applicant reserves the right to address in detail the Examiner's characterizations, conclusions, and rejections in future prosecution.

With respect to the provisional rejections of claims 1-4 under the judicially-created doctrine of obviousness-type double patenting over co-pending and commonly-owned patent application no. 10/306,942 (hereinafter "the '942 application"), Applicant will consider filing a Terminal Disclaimer once the instant application is in condition for allowance or the '942 application issues as a patent. However, Applicant respectfully asserts that, in view of the amendments to the claims and arguments made above, the Examiner is compelled to withdraw the rejections of the claims. Once withdrawn, the only rejection remaining in the instant application would be the provisional obviousness-type double patenting rejection. In view of MPEP § 804 I(B), Applicant respectfully requests that the provisional obviousness-type double patenting rejection be withdrawn and that the instant application be permitted to issue as a patent.

Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C2) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the undersigned attorney of record invites the Examiner to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

Date: September 17, 2010

By: /Erin Nichols Matkaiti/

Erin Nichols Matkaiti
Reg. No. 57,125

PTO/SB/22 (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARMENT OF COMMERCE
Under the paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) **FY 2009** (*Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).*)	Docket Number (Optional) TPO.004.C2
Application Number 11/338,934	Filed January 25, 2006
For Grouping of Image Frames in Video Coding	
Art Unit 2611	Examiner Lee, S.

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

		Fee	Small Entity Fee	
☑	One month (37 CFR 1.17(a)(1))	$130	$65	$ 130.00
☐	Two months (37 CFR 1.17(a)(2))	$490	$245	$
☐	Three months (37 CFR 1.17(a)(3))	$1110	$555	$
☐	Four months (37 CFR 1.17(a)(4))	$1730	$865	$
☐	Five months (37 CFR 1.17(a)(5))	$2350	$1175	$

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☑ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 50-3581.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed (Form PTO/SB/96).

☑ attorney or agent of record. Registration Number 57,125

☐ attorney or agent under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 _____

/Erin Nichols Matkaiti/ — Signature

September 17, 2010 — Date

Erin Nichols Matkaiti — Typed or printed name

952-854-2700 — Telephone Number

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☑ Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.136(a). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 6 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Patent Application Fee Transmittal

Application Number:	11338934
Filing Date:	25-Jan-2006
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Filer:	Erin M. Nichols
Attorney Docket Number:	TPO.004.C2

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 1 month with $0 paid	1251	1	130	130

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**130**

Electronic Acknowledgement Receipt

EFS ID:	8440165
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	17-SEP-2010
Filing Date:	25-JAN-2006
Time Stamp:	11:49:56
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$130
RAM confirmation Number	10189
Deposit Account	503581
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		OAR-TPO004C2-OA25052010.pdf	95492 e2dea1ff0412ff322887c375b928c5a16e435a48	yes	10

Multipart Description/PDF files in .zip description		
Document Description	**Start**	**End**
Amendment/Req. Reconsideration-After Non-Final Reject	1	1
Claims	2	5
Applicant Arguments/Remarks Made in an Amendment	6	10

Warnings:

Information:

2	Extension of Time	ExtofTime-TPO004C2-OA25052010.pdf	312721 4d7d29170f3e25130fbb26d23c6e62c23ed6f1f3	no	2

Warnings:

Information:

3	Fee Worksheet (PTO-875)	fee-info.pdf	30074 f41cf99dda9e5d97013d2a985f9901328a3e8d1f	no	2

Warnings:

Information:

Total Files Size (in bytes):	438287

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
11/338,934	01/25/2006	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT 09/17/2010	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 11	Minus	** 20	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	* 6	Minus	***6	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

AMENDMENT	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/PEGGY YARBOROUGH/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5852630	12/22/1998	Langberg			
	6072831	06/06/2000	Chen			
	6363208	03/26/2002	Nitta et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	2137197C1	9/10/1999	Russia				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	1997	Illgner et al., "Spatially Scalable Video Compression Employing Resolution Pyramids", IEEE Journal on Selected Areas in Communications, Vol. 15, No. 9, December 1, 2007, pg. 1688-1703.
	2000	Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Systems, ISO/IEC 13818-1, Second edition, December 1, 2000, 174 pages.
	5/5/2010	Notice of Allowance dated 5/5/2010 from Russian Application No. 2006110321/09, 11 pages.
	6/17/2010	Office Action dated 6/17/2010 from U.S. Serial No. 11/338,996, 30 pages.
	6/23/2010	Office Action dated 6/23/2010 from U.S. Serial No. 10/306,942, 6 pages.
	5/25/2010	Office Action dated 5/25/2010 from U.S. Serial No. 10/306,942, 6 pages.
	4/09/2009	Office Action dated 4/9/2009 from U.S. Serial No. 10/306,942, 4 pages.
	5/8/2006	Office Action Response dated 5/8/2006 from U.S. Serial No. 10/306,942, 11 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Electronic Patent Application Fee Transmittal

Application Number:	11338934
Filing Date:	25-Jan-2006
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Filer:	Steven R. Funk/Tracey Dotter
Attorney Docket Number:	TPO.004.C2

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
		Total in USD ($)		**180**

Electronic Acknowledgement Receipt

EFS ID:	8190380
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Steven R. Funk
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	10-AUG-2010
Filing Date:	25-JAN-2006
Time Stamp:	17:33:08
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$180
RAM confirmation Number	3997
Deposit Account	503581
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-10-TPO004C2.pdf	55426 3a289480cadfd3eef89a80ef685a2fe46b4ae296	no	2
Warnings:					
Information:					
2	Information Disclosure Statement (IDS) Filed (SB/08)	1449-10-TPO004C2.pdf	53435 5ecde266655871ab0f61200da8cf90c9f89cb2a3	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	FH-TAPA004PA.pdf	999816 e0f3b198e0372dfeee0d292f6afef40141aed93c	no	27
Warnings:					
Information:					
4	NPL Documents	Illgner.pdf	524122 72978d8c3d123a922894913ff66ff592b86674fa	no	16
Warnings:					
Information:					
5	NPL Documents	Informationtechnology.pdf	1641793 d76b603427c9443ca71ff18667af1f1834b69a97	no	174
Warnings:					
Information:					
6	NPL Documents	RUAllowance.pdf	4724947 318f5d65aaeeedf8f2b368397e91b944baced92a	no	11
Warnings:					
Information:					
7	Foreign Reference	RU2137194C1.pdf	973318 31ad8c486895acf06c549d3f83a00eff14e65d24	no	15
Warnings:					
Information:					
8	NPL Documents	TPO004C1-OA06172010.pdf	976305 3669dce1f618145853a25ebde8d2ffbab41c3937	no	30
Warnings:					
Information:					

9	Fee Worksheet (PTO-875)	fee-info.pdf	30215 fa100b9ba2d268b097e5f4bec88273347ac5abcb	no	2

Warnings:	
Information:	
Total Files Size (in bytes):	9979377

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,934	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. §1.97(c)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant respectfully submits the items of information on the enclosed Form 1449 for the attention of the Examiner in the above-identified application.

This statement should be considered because it is submitted after the mailing date of the first Office Action on the merits but before the mailing date of a final Office Action under 37 C.F.R. §1.113 or a Notice of Allowance under 37 C.F.R. §1.311. Please charge Deposit Account 50-3581 (TPO.004.C2) the amount of $180.00 under 37 C.F.R. §1.17(p) for consideration of the item.

In accordance with 37 C.F.R. §1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that the reference is "prior art" within the meaning of 35 U.S.C. §§102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. §1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Steven R. Funk/
Steven R. Funk
Reg. No. 37,830

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,934	01/25/2006	Miska Hannuksela	TPO.004.C2	6610

76385 7590 05/25/2010
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

MAIL DATE	DELIVERY MODE
05/25/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/338,934	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	SIU M. LEE	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 January 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 January 2006* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☒ Certified copies of the priority documents have been received in Application No. 10/306942.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *See Continuation Sheet*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :6/19/2006, 8/18/2006, 11/24/2006, 10/22/2007, 6/16/2008, 3/12/2009, 4/20/2009, 2/08/2010, 2/18/2010, 3/30/2010.

DETAILED ACTION

Claim Objections

1. Claims 1-4, 5-8, and 10-12 are objected to because of the following informalities:

Claim 1, line 4 recites “the previous reference image frame”; there is a lack of antecedent basis for this limitation.

Claims 5, line 4 recites “the previous reference image frame”; there is a lack of antecedent basis for this limitation.

Claim 10, line 3 recites “the previous reference image frame”; there is a lack of antecedent basis for this limitation.

Appropriate correction is required.

Drawings

2. The drawings are objected to because Figure 1 does not have sufficiently descriptive labels. Blank boxes in drawings should be labeled descriptively unless it is a well-known component. For example, the blank box, "100" should be labeled as "Multimedia Source". Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as “amended.” If a drawing figure is to be canceled, the appropriate figure must be removed from the

replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 5-8 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claimed "video encoder" is non-structural per se and because the specification discloses a "software" embodiment (page 5, paragraph 0014 and page 33, paragraph 0097). Therefore, a reasonable interpretation in light of the specification leads to the conclusion that the claim video encoder as pure software (further confirm in claim 9 that computer program code for encoding a compressed video sequence, claim 9, line 7), which does not fall within the definition of a process, machine, manufacture or composition of matter.

Claim 13 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claimed "video decoder" is non-structural per se and because the specification discloses a "software" embodiment (page 5, paragraph 0014 and page 33, paragraph 0097). Therefore, a broadest reasonable interpretation in light of the specification leads to the conclusion that the claim video decoder as pure software (further confirm in claim 9 that computer program code for decoding from the video sequence, claim 14, line 7), which does not fall within the definition of a process, machine, manufacture or composition of matter.

Claim 9 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine, manufacture or composition of matter. The claim is drawn to a "computer program product stored in a computer readable medium". The specification does not define the computer readable medium, therefore, it includes transitory and non-transitory computer readable medium. The broadest reasonable interpretation of the claim in light of the specification concludes that the claim as a whole covers a transitory signal, which does not fall within the definition of a process, machine, manufacture, or composition of matter (In re Nuijten). Therefore, claim 9 does not fall within a statutory category (the rejection can be overcome by specifying the computer readable medium as "non-transitory computer readable medium").

Claim 14 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim is NOT to a process, machine,

manufacture or composition of matter. The claim is drawn to a "computer program product stored in a computer readable medium". The specification does not define the computer readable medium, therefore, it includes transitory and non-transitory computer readable medium. The broadest reasonable interpretation of the claim in light of the specification concludes that the claim as a whole covers a transitory signal, which does not fall within the definition of a process, machine, manufacture, or composition of matter (In re Nuijten). Therefore, claim 9 does not fall within a statutory category (the rejection can be overcome by specifying the computer readable medium as "non-transitory computer readable medium").

Claim 15-16 are directed to a "video signal"; a signal is not limited to that which falls within a statutory category of invention because it is not limit to a process, machine, manufacture, or a composition of matter. Instead, it includes a form of energy. Energy does not fall with in a statutory category since it is clearly not a series of steps or acts to constitute a process, not a mechanical device or combination of mechanical device to constitute a machine, not a tangible physical article or object which is some form of matter to be a product and constitute a manufacture, and not a composite of two or more substances to constitute a composite of matter.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 5-8 are rejected under 112, 1st paragraph as being a single component (video encoder) claim, i.e., where a single component (video encoder) recitation does not appear in combination with another recited component, is subject to an undue breadth rejection. The single component which covered every conceivable ways for achieving the stated purpose was held nonenabling for the scope of claim because the specification disclosed at most only those known to the inventor. See MPEP §2164.08(a).

Claim 13 is rejected under 112, 1st paragraph as being a single component (video decoder) claim, i.e., where a single component (video decoder) recitation does not appear in combination with another recited component, is subject to an undue breadth rejection. The single component which covered every conceivable ways for achieving the stated purpose was held nonenabling for the scope of claim because the specification disclosed at most only those known to the inventor. See MPEP §2164.08(a).

Double Patenting

5. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422

F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

6. Claims 1-2, and 4 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 9 of copending Application No. 10/306942 in view of Kim et al. (US 2003/0063806 A1, hereinafter Kim).

Claim	Instant application	Claim	Co-pending application 10/306942
1	A method for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order, the	9	Claim 1 A method for coding video frames for the purpose of forming a scalable, compressed video sequence comprising video frames coded according to at least a first and a second frame format, **the video frames of the first frame format being independent of the**

	method comprising: encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence.		**other video frames**, and **the video frames of the second frame format being predicted from at least one of the other video frames**, the method comprising encoding, at an encoder, the video sequence as at least part of a first sub-sequence, at least part of which has been formed by coding video frames of the at least first frame format; and encoding, at the encoder, the video sequence as at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format, and **at least one video frame of the second sub-sequence has been predicted from at least one video frame of the first sub-sequence**; and encoding, at the
2	A method according to claim 1, further comprising: encoding the indication into the video sequence as a separate flag included in the header of a slice.		
4	A method according to claim 1, further comprising: encoding into the video sequence an identifier value for the independent sequence.		

			encoder, into the video sequence information indicating which video frames belong to the second sub-sequence. Claim 3 A method according to claim 2, further comprising coding the video sequence into a plural number of scalability layers; and determining the dependency of the video frames of the second sub-sequence **such that at least one video frame of the second sub-sequence is predicted from a group comprising a video frame of a higher scalability layer, a video frame of another sub-sequence in the same scalability layer**. Claim 7 A method according to claim 3, further comprising **determining a**

			unique identifier for each video frame as a combination of layer number, sub-sequence identifier and image number. Claim 9 A method according to claim 7, further comprising **adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence**.

This is a provisional obviousness-type double patenting rejection.

(1) Regarding claim 1:

The co-pending application discloses a method for encoding a video sequence comprising an independent sequence of image frames (video sequence comprising video frame coded according to at least a first and second fame format), wherein at least one reference image frame (second sub-sequence) is predictable from at least one previous image frame that is earlier than a previous reference image frame in decoding order (at least one video frame of the second sub-sequence has been

predicted from at least one video frame of the first sub-sequence or a video frame of another sub-sequence in the same scalability layer), the method comprising:

encoding into the video sequence an indication (identifier) of at least one image frame (determined a unique identifier for each video frame), which is the first image frame, in decoding order, of the independent sequence (since **each** frame has an identifier added to the header field of the video sequence, therefore, it included the first image frame), said at least one reference image frame being included in the sequence (the video sequence includes at least a second sub-sequence, therefore, the claimed at least one reference image frame is included in the video sequence).

Claim 9 of the instant application does not explicitly disclose at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order.

However, Kim discloses an independent video sequence in figure 7 that at least one reference image frame P frame 708) is predictable from at least one previous image frame (I frame 704) that is earlier than a previous reference image frame in decoding order (P frame 706).

It is desirable to have at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order because it reduces latency when decoding the video sequence. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the teaching of Kim in the method of the co-pending application to reduce latency when decoding the video sequence.

(2) Regarding claim 2 (the examiner interprets "a slice" as the video sequence as claim in claim 1):

The copending application discloses:

encoding the indication into the video sequence as a separate flag included in the header of a slice (adding the identifier to the header field of the video sequence or to the header field of the transfer protocol to be used for transferring the video sequence, the examiner interprets the adding process as the encoding process as claim).

(3) Regarding claim 4:

The co-pending application discloses:

encoding into the video sequence an identifier value for the independent sequence (determining a unique identifier for each video frame and adding the identifier to the header field of the video sequence, the examiner interpret adding the unique identifier to the header as encoding the identifier into the video sequence).

7. Claim 3 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 9 of copending Application No. 10/306942 in view of Kim et al. (US 2003/0063806 A1, hereinafter Kim) as apply to claim 1 and further in view of Kawashima et al. (US 4,390,966).

Copending application and Kim discloses all subject matter as discussed in claim 1 and copending application further discloses the encoding identifier values for image frames according to a numbering scheme (layer number, sub-sequence identifier and

image number), but fail to disclose resetting the identifier value for the indicated first image frame of an independent sequence.

However, Kawashima discloses a controller that setting or resetting flags as desired according to user instruction (column 6, lines 37-40).

It is desirable to reset the identifier value for the indicated first image frame of an independent sequence because since the video sequence is independent of other sequences, it would simplify the identifier value and reset the identifier value. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the resetting step as taught by Kawashima in the method of copending application and Kim to simplify the identifier value of the video sequence.

Kim et al. (US 2003/0063806 A1, hereinafter Kim) in view of Ueda (US 6,094,456)

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9. Claims 1 and 5 are rejected under 35 U.S.C. 102(e) as being anticipated by Kim et al. (US 2003/0063806 A1, hereinafter Kim).

Kim discloses a method and encoder (encoder discloses in figure 1B and 1C) for encoding a video sequence comprising an independent sequence of image frames (independent sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising:

encoding into the video sequence an indication of at least one image frame (a video coding apparatus is provided to receive an indication that a first frame is to be intracoded, paragraph 0018, the examiner interprets the first intracoded frame as the claimed indication), which is the first image frame (the first frame is intracoded), in decoding order, of the independent sequence (independent sequence as shown in figure 7), said at least one reference image frame being included in the sequence (the P frame 708 is included in the sequence).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 1, 2, 4, 6, 8, 10, 11, 13, and 15-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kim et al. (US 2003/0063806 A1, hereinafter Kim) in view of Ueda (US 6,094,456).

(1) Regarding claim 1:

Kim discloses a method and encoder (encoder discloses in figure 1B and 1C) for encoding a video sequence comprising an independent sequence of image frames (independent sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising:

encoding into the video sequence at least one image frame (a video coding apparatus is provided to receive an indication that a first frame is to be intracoded, paragraph 0018, the examiner interprets the first intracoded frame as the claimed indication), which is the first image frame (the first frame is intracoded), in decoding order, of the independent sequence (independent sequence as shown in figure 7), said at least one reference image frame being included in the sequence (the P frame 708 is included in the sequence).

Kim fails to explicitly disclose an indication of the first I frame.

However, Ueda discloses an encoded video sequence that comprises a fixed parameter identifier in the header of an input video code that indicates the type of picture from the picture header (I frame, column 10, lines 38-42).

Since Kim discloses the first frame is an I frame, and Ueda discloses having a fixed parameter identifier in the header to indicate whether the frame is an I frame, it is obvious to employ the teaching of Ueda in the encoding method of Kim to include in the header a fixed parameter identifier for identifying the first I frame in order to improve the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the method of Kim to improve the reproduction speed of the encoded video sequence.

(2) Regarding claims 2 and 6:

Kim discloses all subject matter as discussed in claims 1 and 5 but fail to disclose encoding the indication into the video sequence as a separate flag included in the header of a slice.

However, Ueda discloses an encoded video sequence that comprises a fixed parameter identifier in the header of an input video code that indicates the type of picture from the picture header (column 10, lines 38-42).

Since Kim discloses the first frame is an I frame, and Ueda discloses having a fixed parameter identifier in the header to indicate whether the frame is an I frame, it is obvious to employ the teaching of Ueda in the encoding method of Kim to include in the header a fixed parameter identifier for identifying the first I frame in order to improve the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the method of Kim to improve the reproduction speed of the encoded video sequence.

(3) Regarding claims 4 and 8:

Ueda further discloses encoding into the video sequence an identifier value for the independent sequence (an identifier of six bytes indicative of type of picture, column 7, lines 60-64 and column 10, lines 38-42).

(4) Regarding claims 10 and 13:

Kim discloses a method and decoder for decoding a compressed video sequence (sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising

decoding the first two frames of a sequence that yield the exact frame rate, the first two frame as I frame of figure 7 (paragraph 0071) which is the first image frame (I frame 702 in figure 7 is the first frame of the sequence), in decoding order, of an independent sequence, said sequence including the at least one reference image frame (P frame 708 is included in the sequence) and starting the decoding of the video sequence from said first image frame of the independent sequence (as shown in figure 7, the first I frame 702 and the second I frame 704 is for generating the P frame 706-

712, therefore, the decoding processing is start from I frame 702), whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame (as shown in figure 7, the sequence is independent and I frame 702 is the first frame of the sequence and no other frame prior to frame 702 is required).

Kim does not explicitly disclose decoding from the video sequence an indication of at least one image frame which is the first image frame, in decoding order of an independent sequence.

However, Ueda discloses a method that determines from an identifier in the header and determine if the frame is an I frame (step 111 and 112 of figure 12, column 14, lines 31-35).

It is desirable to decoding from the video sequence an indication of at least one image frame which is the first image frame, in decoding order of an independent sequence because it improves the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the decoding method of Kim to improve the reproduction speed of the encoded video sequence.

(5) Regarding claim 11:

Ueda further discloses wherein the indication is a separate flag in the header of a slice (an identifier of six bytes indicative of type of picture, column 7, lines 60-64 and column 10, lines 38-42).

(6) Regarding claim 15:

Kim discloses a signal comprising a compressed video sequence including an independent sequence of image frames (sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising:

at least one image frame (the first frame is an intracoded frame), which is the first image frame, in decoding order, of the independent sequence (the first I frame 702 is the first frame of the sequence as shown in figure 7), said at least one reference image frame being included in the sequence (P frame 708 is included in the sequence as shown in figure 7).

Kim fails to explicitly disclose an indication of the first I frame.

However, Ueda discloses an encoded video sequence that comprises a fixed parameter identifier in the header of an input video code that indicates the type of picture from the picture header (I frame, column 10, lines 38-42).

Since Kim discloses the first frame is an I frame, and Ueda discloses having a fixed parameter identifier in the header to indicate whether the frame is an I frame, it is obvious to employ the teaching of Ueda in the encoding method of Kim to include in the header a fixed parameter identifier for identifying the first I frame in order to improve the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the method of Kim to improve the reproduction speed of the encoded video sequence.

(7) Regarding claim 16:

Ueda further discloses encoding into the video sequence an identifier value for the independent sequence (an identifier of six bytes indicative of type of picture, column 7, lines 60-64 and column 10, lines 38-42).

12. Claims 3 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kim et al. (US 2003/0063806 A1, hereinafter Kim) in view of Ueda (US 6,094,456) and Kawashima et al. (US 4,390,966).

Kim discloses all subject matter as discussed in claims 1 and 5 except (a) encoding identifier values for image frames according to a numbering scheme, and (b) resetting the identifier value for the indicated first image frame of an independent sequence.

With respect to (a), Ueda discloses a fixed parameter identifier of six bits in the header of an input video code that indicates the type of picture from the picture header (column 7, lines 60-64 and column 10, lines 38-42, the examiner interprets the bits as a representation of numbering scheme in binary).

Since Kim discloses the first frame is an I frame, and Ueda discloses having a fixed parameter identifier in the header to indicate whether the frame is an I frame, it is

obvious to employ the teaching of Ueda in the encoding method of Kim to include in the header a fixed parameter identifier for identifying the first I frame in order to improve the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the method of Kim to improve the reproduction speed of the encoded video sequence.

With respect to (b), Kawashima discloses a controller that setting or resetting flags as desired according to user instruction (column 6, lines 37-40).

It is desirable to reset the identifier value for the indicated first image frame of an independent sequence because since the video sequence is independent of other sequences, it would simplify the identifier value and reset the identifier value. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to employ the resetting step as taught by Kawashima in the method of copending application and Kim to simplify the identifier value of the video sequence.

13. Claims 9 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kim et al. (US 2003/0063806 A1, hereinafter Kim) in view of Langberg et al. (US 5,852,630).

(1) Regarding claim 9:

Kim discloses a method and encoder (encoder discloses in figure 1B and 1C) for encoding a video sequence comprising an independent sequence of image frames

(independent sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising:

encoding into the video sequence an indication of at least one image frame (a video coding apparatus is provided to receive an indication that a first frame is to be intracoded, paragraph 0018, the examiner interprets the first intracoded frame as the claimed indication), which is the first image frame (the first frame is intracoded), in decoding order, of the independent sequence (independent sequence as shown in figure 7), said at least one reference image frame being included in the sequence (the P frame 708 is included in the sequence).

Kim fails to disclose implementing the encoder by a computer program product, stored on a computer readable medium and executable in a data processing unit.

However, Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Kim would have been implemented in a software. The implemented software would perform same function of the hardware for less expense, adaptability,

and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg to implement the method and encoder of Kim in order to reduce cost and improve the adaptability and flexibility of the communication system.

(2) Regarding claim 14:

Kim discloses a method and decoder for decoding a compressed video sequence (sequence as shown in figure 7), wherein at least one reference image frame (as there is no definition of a reference frame, the examiner interprets a P frame as a reference frame, P frame 708 in figure 7) is predictable from at least one previous image frame (I frame 704 in figure 7) that is earlier than the previous reference image frame (P frame 706 in figure 7) in decoding order, the method comprising

decoding the first two frames of a sequence that yield the exact frame rate, the first two frame as I frame of figure 7 (paragraph 0071) which is the first image frame (I frame 702 in figure 7 is the first frame of the sequence), in decoding order, of an independent sequence, said sequence including the at least one reference image frame (P frame 708 is included in the sequence) and starting the decoding of the video sequence from said first image frame of the independent sequence (as shown in figure 7, the first I frame 702 and the second I frame 704 is for generating the P frame 706-712, therefore, the decoding processing is start from I frame 702), whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame (as shown in figure 7, the sequence is independent and I frame 702 is the first frame of the sequence and no other frame prior to frame 702 is required).

Kim does not explicitly disclose (a) decoding from the video sequence an indication of at least one image frame which is the first image frame, in decoding order of an independent sequence, and (b) implementing the method by a computer program product, stored on a computer readable medium and executable in a data processing unit.

With respect to (a), Ueda discloses a method that determines from an identifier in the header and determine if the frame is a I frame (step 111 and 112 of figure 12, column 14, lines 31-35).

It is desirable to decoding from the video sequence an indication of at least one image frame which is the first image frame, in decoding order of an independent sequence because it improves the reproduction speed (column 2, lines 24-27).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to include in the header a fixed parameter identifier for identifying the first I frame as taught by Ueda in the decoding method of Kim to improve the reproduction speed of the encoded video sequence.

With respect to (b), Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system or method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Kim would have been implemented in a software. The implemented

software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg to implement the method and encoder of Kim and Ueda in order to reduce cost and improve the adaptability and flexibility of the communication system.

14. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kim et al. (US 2003/0063806 A1, hereinafter Kim) in view of Ueda (US 6,094,456) as applied to claim 10 and further in vie of Kawashima et al. (US 4,390,966).

Kim and Ueda disclose all subject matter as discussed in claim 10, and Ueda further discloses a fixed parameter identifier of six bits in the header of an input video code that indicates the type of picture from the picture header (column 7, lines 60-64 and column 10, lines 38-42, the examiner interprets the bits as a representation of numbering scheme in binary).

Kim and Ueda fail to disclose resetting the identifier value for the indicated first image frame of an independent sequence.

However, Kawashima discloses a controller that setting or resetting flags as desired according to user instruction (column 6, lines 37-40).

It is desirable to reset the identifier value for the indicated first image frame of an independent sequence because since the video sequence is independent of other sequences, it would simplify the identifier value and reset the identifier value. Therefore, it would have been obvious to one of ordinary skill in the art at the time of

invention to employ the resetting step as taught by Kawashima in the method of copending application and Kim to simplify the identifier value of the video sequence.

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Cuccia (US 202/0051621 A1) discloses an instant replay on digital video optimized using non MPEG frame tag.

Hannuksela et al. (US 2001/0040700 A1) discloses a video coding.

Apostolopoulos (US 7,103,669 B2) discloses a video communication method and system employing multiple state encoding and path diversity.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU M. LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 7:30-4:00 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Examiner, Art Unit 2611
5/21/2010

/CHIEH M FAN/

Supervisory Patent Examiner, Art Unit 2611

Notice of References Cited	Application/Control No. 11/338,934	Applicant(s)/Patent Under Reexamination HANNUKSELA, MISKA	
	Examiner SIU M. LEE	Art Unit 2611	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,094,456	07-2000	Ueda, Hiroaki	375/240.12
*	B	US-2003/0063806	04-2003	Kim et al.	382/236
*	C	US-4,390,966	06-1983	Kawashima et al.	712/229
*	D	US-7,103,669	09-2006	Apostolopoulos, John G.	709/231
*	E	US-2001/0040700	11-2001	Hannuksela et al.	358/261.2
*	F	US-2002/0051621	05-2002	Cuccia, David W.	386/68
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5122875	06/16/1992	Raychaudhuri et al.			
	5838265	11/17/1998	Adolph			
	5877812	03/02/1999	Krause et al.			
	6314139	11/06/2001	Koto et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	02/04/2010	Copy of Allowance Notification with translation dated 2/4/2010 from parallel Russian Application No. 2006110322, 17 pages.
	02/25/2010	Copy of translated Office Action dated 2/25/2010 from parallel Japanese Application No. 2006-120297, 5 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Electronic Acknowledgement Receipt

EFS ID:	7316644
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Steven R. Funk
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	30-MAR-2010
Filing Date:	25-JAN-2006
Time Stamp:	15:45:16
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-9-TPO004C2.pdf	54555 9dff5f35f68114ec803240b8368ae5a9e9d88a1b	no	2

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	1449-9-TPO004C2.pdf	52392 2b0412da4c73a2585f912f36d31dc7601983ba3e	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	OA20100225.pdf	98113 56c0b5dbea6eaa02a0dd55397547b2bad6afd76f	no	5
Warnings:					
Information:					
4	NPL Documents	RUAllowance.pdf	2428420 81a2a0594f6e1200fcb4ddfe9abda3436c7a50f9	no	17
Warnings:					
Information:					
		Total Files Size (in bytes):	2633480		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,934	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of the foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a

reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Steven R. Funk/
Steven R. Funk
Reg. No. 37,830

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	6496980	12/17/2002	Tillman et al.			
	6510553	01/21/2003	Hazra			
	6639943	10/28/2003	Radha et al.			
	6614936	09/02/2003	Wu et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	EP0898427	2/24/1999	EP			NA	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	5/7/1996	Elloumi et al., "Issues in Multimedia Scaling: the MPEG Video Streams Case", Emerging Technologies and Applications in Communications, IEEE Computer Society, May 7, 1996.
	2/1/2002	Hannuksela, "Enhanced Concept of GOP", Joint Video Team of ISO-IES MPEG 7 itu-T VCEG, 2/1/2002.
	10/12/1997	Soderquist et al., "Memory Traffic and Data Cache Behavior of an MPEG-2 Software Decoder", IEEE Computer Society, October 12, 1997.
	2/2/2010	Copy of Office Action of related European application no. 06101013.8 dated 02/02/2010.
	2/2/2010	Copy of Office Action of related European application no. 06101014.6 dated 02/02/2010.
	3/16/2006	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 3/16/2006, 4 pages.
	8/10/2006	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 8/10/2006, 16 pages.
	9/11/2007	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 9/11/2007, 4 pages.
	3/13/2007	Copy of Office Action of parallel U.S. Application No. 10/306,942 dated 3/13/2007, 7 pages.
	12/7/2009	Copy of Notice of Allowance of parallel U.S. Application No. 10/306,942 dated 12/7/2009, 8 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

	5/6/2006	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 8/6/2006, 13 pages.
	11/9/2006	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 11/9/2006, 16 pages.
	5/31/2007	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 5/31/2007, 12 pages.
	7/9/2009	Copy of Office Action Response of parallel U.S. Application No. 10/306,942 dated 7/9/2009, 12 pages.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

Electronic Acknowledgement Receipt

EFS ID:	7035879
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Steven R. Funk
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	18-FEB-2010
Filing Date:	25-JAN-2006
Time Stamp:	19:11:49
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-8-TPO004C2.pdf	54738 440dc377c432bfd5e70fff76ddca3c77468add40	no	2

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	1449-8-TPO004C2.pdf	62014 7256dd08f9e1ff4b228f58983b044c089b45b2ad	no	2
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	Elloumi.pdf	337636 7a38047d06aa5d3f7cd8e9e602921ea5c5b034ff	no	4
Warnings:					
Information:					
4	NPL Documents	Hannuksela.pdf	712317 50ee279cd1a1b1c7551108ba7cbed2abfb2c5507	no	14
Warnings:					
Information:					
5	NPL Documents	Soderquist.pdf	641172 08a2996ec505864cfaa3f7b8fa668d9f73f3a2ca	no	6
Warnings:					
Information:					
6	NPL Documents	EPSR-1.pdf	1448766 ac197f5e58085b037546485233f2c1327fd0f075	no	5
Warnings:					
Information:					
7	NPL Documents	EPSR-2.pdf	1732473 a0aa0d91756e7d74226ada4a85633e6f3969b1a9	no	5
Warnings:					
Information:					
8	Foreign Reference	EP0898427.pdf	2658400 a4a37ba6cebd0b966345da06aea44c900df3acab	no	60
Warnings:					
Information:					
9	NPL Documents	TAPA004PA-Allowance.pdf	507388 8687fa54e8edcd1b0f4307d369d6574904ccf80e	no	8
Warnings:					
Information:					
10	NPL Documents	TAPA004PA-OA09042009.pdf	328025 ee2eb11396b0d32275b729a41dcab7d02c43f190	no	7

Warnings:					
Information:					
11	NPL Documents	TAPA004PA-OA10082006.pdf	763601 b767b90add0d470accc93e5830f1b4af899fd807	no	16
Warnings:					
Information:					
12	NPL Documents	TAPA004PA-OA11092007.pdf	160508 fe1556ebcf37a1e60a1f35d6e46d29b7f2fe0fcd	no	4
Warnings:					
Information:					
13	NPL Documents	TAPA004PA-OA13032007.pdf	240311 b4f05eecd63d96dcc100f84be33ae6d606179a02	no	5
Warnings:					
Information:					
14	NPL Documents	TAPA004PA-OA16032006.pdf	169633 a6390f80af78ba4d9b79d8573a0af449252ae4bb	no	4
Warnings:					
Information:					
15	NPL Documents	TAPA004PA-OAR10082006.pdf	808657 c424432fc9383467ba6351f18e30eff8eac8e64d	no	16
Warnings:					
Information:					
16	NPL Documents	TAPA004PA-OAR-09042009.pdf	535372 2f1206b68db40e4ca8daad28ab010b904e168d6a	no	12
Warnings:					
Information:					
17	NPL Documents	TAPA004PA-OAR-13032007.pdf	543295 a8a2484b1702714bbea0af5d0d0a916eda91a5b7	no	12
Warnings:					
Information:					
18	NPL Documents	TAPA004PA-RR16032006.pdf	589717 dd9806cb4b2d7d65b32b33c965fa551ab50f97cb	no	13
Warnings:					
Information:					
			Total Files Size (in bytes):	12294023	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,934	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of the foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided. Also attached is a copy of the English translation of the Japanese Office Action.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to

establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Steven R. Funk/
Steven R. Funk
Reg. No. 37,830

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	11-177987	02.07.1999	JP			Abstract	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	2010	English Translation of the Office Action of parallel Japanese application no. 2006-120296 dated 01/06/2010.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Electronic Acknowledgement Receipt

EFS ID:	6928529
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Steven R. Funk
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	08-FEB-2010
Filing Date:	25-JAN-2006
Time Stamp:	10:26:33
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-7-TPO004C2.pdf	55083 df08ecbfb1ca848316b677327a649e6e49a99e4c	no	2

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	1449-7-TPO004C2.pdf	55900 7c63015565e6ffd90e77f1555c5328f8acfeb6d0	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Foreign Reference	JP11177987.pdf	816939 84f6ab2c6bf4de5f13d25754972bd8ee4e104810	no	15
Warnings:					
Information:					
4	NPL Documents	20100106-JPOA.pdf	64310 4f9278cb198536d67dee9192adfefa17c1944f59	no	3
Warnings:					
Information:					
		Total Files Size (in bytes):	992232		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,934	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

The Japanese reference was first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of the foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided. Also attached is a copy of the English translation of the Japanese Office Action.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.854.2700

By: /Steven R. Funk/
Steven R. Funk
Reg. No. 37,830

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	6,26,158	7/24/2001	Hata et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	10-271511	10/09/1998	JP				X

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Electronic Acknowledgement Receipt

EFS ID:	5179746
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C2
Receipt Date:	20-APR-2009
Filing Date:	25-JAN-2006
Time Stamp:	16:27:02
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Foreign Reference	JP10271511.pdf	1504119 30117798bbfb84b3a6032e8f31dcb02fa3fee47f	no	36

Warnings:	
Information:	

2	Transmittal Letter	IDS-6-TPO004C2.pdf	55984 8475d972cde663804faa27ebac7e402f8151d1ce	no	2
Warnings:					
Information:					
3	Information Disclosure Statement (IDS) Filed (SB/08)	Form1449supp6-TPO004C2.pdf	55814 d995fc2cc09eb5bb5a27f768a3556f275310f1a2	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
4	NPL Documents	JP-OfficeAction-26032009.pdf	185000 3510bcc0224e2113b3825c248487950a40000166	no	6
Warnings:					
Information:					
5	NPL Documents	JP-OfficeActionEngligh-26032009pdf.pdf	189167 c8af3ce98c9773d06b6f6a1077e25746014439df	no	9
Warnings:					
Information:					
		Total Files Size (in bytes):	1990084		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,934	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

The Japanese reference was first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), and the 05 August 2003 Official Gazette Notice, only a copy of the foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided. The U.S. patent (No. 6,266,158) is included as an English-language version of the cited Japanese reference. Also attached is a copy of the Japanese Office Action with an English translation thereof.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

By: /Erin M. Nichols/
Erin M. Nichols
Reg. No. 57,125

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,934	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Confirmation No.:	6610	Customer No.:	76385
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being transmitted to the U.S. Patent and Trademark Office via the Office electronic filing system in accordance with 37 CFR 1.6(a)(4) on March 12, 2009.

By: /Tracey M. Dotter/
Tracey M. Dotter

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

These items were also first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), copies of the foreign references listed on the enclosed Form 1449 are provided. As indicated in the communication from the Chinese patent office (a copy with an English translation is attached) the English-language EP publication corresponds to the cited Chinese reference.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: March 12, 2009

By: /Erin M. Nichols/
Erin M. Nichols
Reg. No. 57,125

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	EP 0798932	10/1/1997	EP			NA	
	CN 1173783A	2/18/1998	China				X

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

Electronic Acknowledgement Receipt

EFS ID:	4945520
Application Number:	11338934
International Application Number:	
Confirmation Number:	6610
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	76385
Filer:	Erin M. Nichols/Tracey Dotter
Filer Authorized By:	Erin M. Nichols
Attorney Docket Number:	TPO.004.C2
Receipt Date:	12-MAR-2009
Filing Date:	25-JAN-2006
Time Stamp:	12:21:57
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement Letter	IDS-5-TPO004C2.pdf	56138 998785e1d84e98bdca6f61f97d7a14ae6f0c0a16	no	2

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	1449-5-TPO004C2.pdf	55772 17cd1e2281a4e7c7994d6617c0338ffa16f60597	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	ChineseOfficeAction-TPO004C1.pdf	577884 a243ee21afd09a7477edd47fbf63fa7d7e8dafe3	no	9
Warnings:					
Information:					
4	Foreign Reference	CN1173783A.pdf	1780503 2bbcb9681463a00d81a20d40bac1322679baf26c	no	36
Warnings:					
Information:					
5	Foreign Reference	EP0798932.pdf	1101670 d7b7193a23b7a6f4e874c79af8c28b44b24b306d	no	22
Warnings:					
Information:					
		Total Files Size (in bytes):	3571967		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
11/338,934		2621	

OC000000032242965

Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 76385 on 09/23/2008

- **Correspondence Address**
- **Power of Attorney Address**

The address of record for Customer Number 76385 is:

76385
Hollingsworth & Funk, LLC
8009 34th Avenue South
Suite 125
Minneapolis, MN 54425

OIPE JUN 16 2008 PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,934	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on June 10, 2008.

By: Rennae Johnson

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the item of information listed on the enclosed Form 1449 is brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the item listed on the enclosed Form 1449.

This item was also first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.

Applicant has not provided a copy of the U.S. publication/patent listed on the enclosed Form 1449 in accordance with 37 C.F.R. §1.98(a)(2).

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to

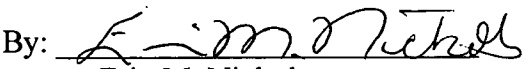
establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: June 10, 2008

By: 
Erin M. Nichols
Reg. No. 57,125

Date Mailed: June 10, 2008

FORM 1449*	INFORMATION DISCLOSURE STATEMENT	Docket Number: TPO.004.C2	Application Number: 11/338,934
	IN AN APPLICATION	Applicant: HANNUKSELA	
	(Use several sheets if necessary)	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

OIPE IAP76 JUN 16 2008 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	20010024472	09/27/2001	Sporer et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

OIPE 1497
OCT 22 2007
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,934	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on October 17, 2007.

By: Rennae Johnson
Rennae Johnson

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. § 1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted before the mailing date of a first Office Action on the merits for the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

These items were also first cited in a communication from a foreign patent office regarding a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached document do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

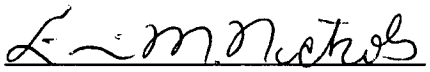
No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. § 1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP § 609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: October 17, 2007

By: 
Erin M. Nichols
Reg. No. 57,125

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION	Docket Number: TAPA.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: Jan. 25, 2006	Group Art Unit: 2613

OIPE IAP17 OCT 22 2007 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5,144,426	09.01.1992	Tanaka et al.			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	01/84850	8 Nov 2001	WO			NA	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

OIPE
NOV 24 2006
IAP43
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,934	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on November 21, 2006.

By: Rennae Johnson

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. §1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the item of information listed on the enclosed Form 1449 is brought to the attention of the Examiner.

This statement should be considered because it is submitted either within three months of the filing date or before the first Office Action on the merits of the above-identified application. Accordingly, no fee is due for consideration of the item listed on the enclosed Form 1449.

In accordance with 37 C.F.R. §1.98(a)(2), and the 5 August 2003 Official Gazette Notice, only a copy of each foreign document or other non-patent published document listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached document do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

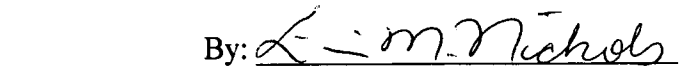
No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. §1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the item listed is respectfully requested. Pursuant to the provisions of MPEP §609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: November 20, 2006

By: 
Erin M. Nichols
Reg. No. 57,125

OIPE IAP43 NOV 24 2006 PATENT & TRADEMARK OFFICE

FORM 1449*	INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
		Applicant: HANNUKSELA	
		Filing Date: 1/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	RU 2 123 769	20.12.1998	Russia			Abstract Only	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

OIPE
AUG 18 2006
PATENT & TRADEMARK OFFICE

IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,934	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

OIPE
AUG 18 2006
PATENT & TRADEMARK OFFICE

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on August 15, 2006.

By: Tracey M. Dotter
Tracey M. Dotter

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT (37 C.F.R. §1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the item of information listed on the enclosed Form 1449 is brought to the attention of the Examiner.

This statement should be considered because it is submitted either within three months of the filing date or before the first Office Action on the merits of the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

In accordance with 37 C.F.R. §1.98(a)(2), and the 5 August 2003 Official Gazette Notice, only a copy of each foreign document or other non-patent published document listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. §1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP §609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: August 15, 2006

By: E. M. Nichols
Erin M. Nichols
Reg. No. 57,125

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 1/25/2006	Group Art Unit: 2613

OIPE AUG 18 2006 IAP19 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS						
EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS							
	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)		
	1998	Wenger, "Temporal Scalability Using P-pictures for Low-latency Applications", IEEE Workshop, December 1998, pp. 559-564.

EXAMINER	DATE CONSIDERED
EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.	

OIPE IAP2
JUN 19 2006
PATENT & TRADEMARK OFFICE

IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Unassigned
Serial No.:	11/338,934	Group Art Unit:	2613
Filed:	January 25, 2006	Docket No.:	TPO.004.C2
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on June 13, 2006.

By Rennae Johnson
Rennae Johnson

INFORMATION DISCLOSURE STATEMENT (37 C.F.R. §1.97(b))

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

With regard to the above-identified application, the items of information listed on the enclosed Form 1449 are brought to the attention of the Examiner.

This statement should be considered because it is submitted either within three months of the filing date or before the first Office Action of the above-identified application. Accordingly, no fee is due for consideration of the items listed on the enclosed Form 1449.

In accordance with 37 C.F.R. §1.98(a)(2), and the 5 August 2003 Official Gazette Notice, only a copy of each foreign document or other non-patent published document listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

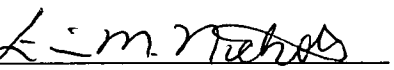
No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§102 and 103 and Applicant reserves the right, pursuant to 37 C.F.R. §1.131 or otherwise, to establish that the reference(s) are not "prior art." Moreover, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended.

Consideration of the items listed is respectfully requested. Pursuant to the provisions of MPEP §609, it is requested that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: June 13, 2006

By: 
Erin M. Nichols
Reg. No. 57,125

OIPE IAP2 JUN 19 2006 PATENT & TRADEMARK OFFICE

FORM 1449* INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C2	Application Number: 11/338,934
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2613

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5,699,476	16.12.1997	VAN DER MEER			
	5,774,593	30.6.1998	ZICK ET AL.			
	6,108,382	22.8.2000	GRINGERI ET AL.			
	6,307,886	23.10.2001	WESTERMANN			
	6,337,881	8.1.2002	CHADDHA			

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
	WO 00/67469	9.11.2000	PCT			n/a	
	WO 01/47283	28.06.2001	PCT			n/a	
	WO 02/054776	11.7.2002	PCT			n/a	
	EP 1 005 232	31.05.2000	EP			n/a	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	9-17 Oct. 2002	Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, *Editor's Proposed Draft Text Modification for Joint Video Specification (ITU-T Rec. 264 ISO/IEC 14496-10 AVC)*, Geneva modifications draft 37.
	15 Jan. 1998	ITU-Telecommunications Standardization Union, *Draft Text of Recommendation H.263 Version 2 ("H.263+") for Decision.*
	7/10/02	ITU-Telecommunications Standardization Union, *H.26L Test Model Long Term Number 8 (TML-8) Draft0*, Document VCEG-N10. pp. 1-46.
	Nov. 2000	ITU-Telecommunications Standardization Union, *Draft for "H.263++" Annexes U,V, and W to Recommendation H.263.*
	12 Nov. 2002	David Singer and Toby Walker, *Study Text of ISO/IEC 14496-15/FCD,* pp. 1-25.
	12 Nov. 2002	David Singer and Toby Walker, *Study Text of ISO/IEC 14496-15/FCD (Revised), 26* pages.
	Sept. 2002	Hannuksela, M et al.; Sub-Picture: ROI Coding and Unequal Error Protection, IEEE 2002 Conference, September 2002, USA (internet publication).
	Sept. 2002	Hannuksela, M et al.; Sub-Picture: Video-Coding for Unequal Error Protection, XI European Signal Processing Conference, September 2002, Toulouse, France (internet publication).
	7.12.2001	Wang, Y.K. et al.; Core Experiment Description of Sub-Picture Coding, ITU 15. Meeting 3-7.12.2001, Pattaya, Thailand (internet publication)
	2001	B.A. Lokshin, "Digital Broadcasting: from Studio to Audience" Cyrus Systems, Moscow, 2001. (translation included)

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449) Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

17224 U.S. PTO
012506

113260 U.S.PTO
11/338934
012506

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: HANNUKSELA

Docket: TPO.004.C2

Title: GROUPING OF IMAGE FRAMES IN VIDEO CODING

CERTIFICATE UNDER 37 CFR 1.10:
"Express Mail" mailing label number: EV 712609393 US
Date of Deposit: January 25, 2006
I hereby certify that this paper or fee is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

By: Tracey M. Dotter
Tracey M. Dotter

CONTINUATION PATENT APPLICATION UNDER 37 C.F.R. § 1.53(b)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This is a request for filing a continuation patent application under 37 CFR § 1.53(b) of pending patent application serial number 10/306,942 filed on November 29, 2002, which claims priority under 35 U.S.C. §119 from corresponding Finnish patent application No. 20020127, filed in Finland on January 23, 2002.

We are transmitting herewith the following:

☒ Transmittal sheet containing Certificate under 37 CFR 1.10 (2 pages).
☒ Patent Application: Pages numbered 1-38, with 1pg. Abstract; 16 claims.
☒ 4 sheets of formal drawings, having Figures 1-2, 3A-3B, 4A-4C, 5A-5C, 6A-6C, and 7.
☒ Copy of the executed Declaration from the priority application (3 pages).
☒ A copy of the Change of Correspondence Address submitted in the prior pending application. Please send all future correspondence to the following address:

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

☒ The priority pending application is assigned to: Nokia Corporation at Reel/Frame No.: 013741/0536.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: HANNUKSELA

Docket: TPO.004.C1

Title: GROUPING OF IMAGE FRAMES IN VIDEO CODING

☒ The Power of Attorney in the prior pending application is to: Steven R. Funk, Reg. No. 37,830.

☒ Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C2) in the amount of $1,800.00 for the filing fees. Authorization is also given to charge/credit Deposit Account No. 50-3581 (TPO.004.C2) for any deficiencies/overpayments.

☒ 1 Return Postcard

The fees are calculated as follows:

Basic Filing Fee =	$ 300.00
Search Fee =	$ 500.00
Examination Fee =	$ 200.00
Fee for total # of claims over 20 = $50 * (16 – 20) =	$ 0.00
Fee for total # of independent claims over 3 = $200 * (7 – 3) =	$ 800.00
TOTAL FILING FEES DUE =	$1,800.00

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: January 25, 2006

By: _____
Steven R. Funk
Reg. No. 37,830

OIPE JC48
AUG 17 2005
PATENT & TRADEMARK OFFICE

PATENT APPLICATION

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	Miska Hannuksela	Examiner:	Unassigned
Serial No.:	10/306,942	Group Art Unit:	2125
Filed:	11/29/2002	**New Docket No.:**	**TAPA.004PA**
		Old Docket No.:	KOLS.021PA

CERTIFICATE UNDER 37 CFR 1.8: The undersigne d hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, United States Patent and Trademark O ffice, P.O. Box 1450, Alexandria, VA 22313-1450, on August 15, 2005.

By: _____

CHANGE OF CORRESPONDENCE ADDRESS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir or Madam:

Please update the attorney docket number as reflected above (**TAPA.004PA**), and change the correspondence Address for the above-referenced application as follows:

Hollingsworth & Funk, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
(952) 854-2700

If there are any questions regarding this matter, please call the undersigned at 952-854-2700.

Respectfully submitted,

By: _____
Steven R. Funk
Reg. No.: 37,830

17224 U.S. PTO
012506

113260 U.S.PTO
11/338934
012506

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: HANNUKSELA

Docket: TPO.004.C2

Title: GROUPING OF IMAGE FRAMES IN VIDEO CODING

CERTIFICATE UNDER 37 CFR 1.10:
"Express Mail" mailing label number: EV 712609393 US
Date of Deposit: January 25, 2006
I hereby certify that this paper or fee is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

By: Tracey M. Dotter
Tracey M. Dotter

CONTINUATION PATENT APPLICATION UNDER 37 C.F.R. § 1.53(b)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This is a request for filing a continuation patent application under 37 CFR § 1.53(b) of pending patent application serial number 10/306,942 filed on November 29, 2002, which claims priority under 35 U.S.C. §119 from corresponding Finnish patent application No. 20020127, filed in Finland on January 23, 2002.

We are transmitting herewith the following:

☒ Transmittal sheet containing Certificate under 37 CFR 1.10 (2 pages).
☒ Patent Application: Pages numbered 1-38, with 1pg. Abstract; 16 claims.
☒ 4 sheets of formal drawings, having Figures 1-2, 3A-3B, 4A-4C, 5A-5C, 6A-6C, and 7.
☒ Copy of the executed Declaration from the priority application (3 pages).
☒ A copy of the Change of Correspondence Address submitted in the prior pending application. Please send all future correspondence to the following address:

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

☒ The priority pending application is assigned to: Nokia Corporation at Reel/Frame No.: 013741/0536.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: HANNUKSELA

Docket: TPO.004.C1

Title: GROUPING OF IMAGE FRAMES IN VIDEO CODING

☒ The Power of Attorney in the prior pending application is to: Steven R. Funk, Reg. No. 37,830.

☒ Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C2) in the amount of $1,800.00 for the filing fees. Authorization is also given to charge/credit Deposit Account No. 50-3581 (TPO.004.C2) for any deficiencies/overpayments.

☒ 1 Return Postcard

The fees are calculated as follows:

Basic Filing Fee =	$ 300.00
Search Fee =	$ 500.00
Examination Fee =	$ 200.00
Fee for total # of claims over 20 = $50 * (16 – 20) =	$ 0.00
Fee for total # of independent claims over 3 = $200 * (7 – 3) =	$ 800.00
TOTAL FILING FEES DUE =	$1,800.00

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
952.854.2700

Date: January 25, 2006

By: _____
Steven R. Funk
Reg. No. 37,830

OIPE JC48
AUG 17 2005
PATENT & TRADEMARK OFFICE

PATENT APPLICATION

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	Miska Hannuksela	Examiner:	Unassigned
Serial No.:	10/306,942	Group Art Unit:	2125
Filed:	11/29/2002	**New Docket No.:**	**TAPA.004PA**
		Old Docket No.:	KOLS.021PA

CERTIFICATE UNDER 37 CFR 1.8: The undersigne d hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Commissioner for Patents, United States Patent and Trademark O ffice, P.O. Box 1450, Alexandria, VA 22313-1450, on August 15, 2005.

By: _____

CHANGE OF CORRESPONDENCE ADDRESS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir or Madam:

Please update the attorney docket number as reflected above (**TAPA.004PA**), and change the correspondence Address for the above-referenced application as follows:

Hollingsworth & Funk, LLC
8009 34th Avenue South, Suite 125
Minneapolis, MN 55425
(952) 854-2700

If there are any questions regarding this matter, please call the undersigned at 952-854-2700.

Respectfully submitted,

By: _____
Steven R. Funk
Reg. No.: 37,830

GROUPING OF IMAGE FRAMES IN VIDEO CODING

RELATED PATENT DOCUMENTS

[0001] This application is a continuation of U.S. Patent Application Serial No. 10/306,942 filed on November 29, 2002 which is incorporated herein by reference in its entirety.

FIELD OF THE INVENTION

[0002] The invention relates to the grouping of multimedia files, particularly video files and particularly in connection with streaming.

BACKGROUND OF THE INVENTION

[0003] The term 'streaming' refers to simultaneous sending and playback of data, typically multimedia data, such as audio and video files, in which the recipient may begin data playback already before all the data to be transmitted has been received. Multimedia data streaming systems comprise a streaming server and terminal devices that the recipients use for setting up a data connection, typically via a telecommunications network, to the streaming server. From the streaming server the recipients retrieve either stored or real-time multimedia data, and the playback of the multimedia data can then begin, most advantageously almost in real-time with the transmission of the data, by means of a streaming application included in the terminal.

[0004] From the point of view of the streaming server, the streaming may be carried out either as normal streaming or as progressive downloading to the terminal. In normal streaming the transmission of the multimedia data and/or the data contents are controlled either by making sure that the bit rate of the transmission substantially corresponds to the playback rate of the terminal device, or, if the telecommunications network used in the transmission causes a bottleneck in data transfer, by making sure that the bit rate of the transmission substantially corresponds to the bandwidth available in the telecommunications network. In progressive downloading the transmission of the multimedia data and/or the data contents do not necessarily have to be interfered with at all, but the multimedia files are transmitted as such to the recipient, typically by using transfer protocol flow control. The terminals then receive, store and reproduce an exact copy of the data transmitted from the server, which copy can then be later reproduced again on the terminal without

needing to start a streaming again via the telecommunications network. The multimedia files stored in the terminal are, however, typically very large and their transfer to the terminal is time-consuming, and they require a significant amount of storage memory capacity, which is why a normal streaming is often preferred.

[0005] The video files in multimedia files comprise a great number of still image frames, which are displayed rapidly in succession (of typically 15 to 30 frames per s) to create an impression of a moving image. The image frames typically comprise a number of stationary background objects, determined by image information which remains substantially unchanged, and few moving objects, determined by image information that changes to some extent. The information comprised by consecutively displayed image frames is typically largely similar, i.e. successive image frames comprise a considerable amount of redundancy. The redundancy appearing in video files can be divided into spatial, temporal and spectral redundancy. Spatial redundancy refers to the mutual correlation of adjacent image pixels, temporal redundancy refers to the changes taking place in specific image objects in subsequent frames, and spectral redundancy to the correlation of different colour components within an image frame.

[0006] To reduce the amount of data in video files, the image data can be compressed into a smaller form by reducing the amount of redundant information in the image frames. In addition, while encoding, most of the currently used video encoders downgrade image quality in image frame sections that are less important in the video information. Further, many video coding methods allow redundancy in a bit stream coded from image data to be reduced by efficient, lossless coding of compression parameters known as VLC (Variable Length Coding).

[0007] In addition, many video coding methods make use of the above-described temporal redundancy of successive image frames. In that case a method known as motion-compensated temporal prediction is used, i.e. the contents of some (typically most) of the image frames in a video sequence are predicted from other frames in the sequence by tracking changes in specific objects or areas in successive image frames. A video sequence always comprises some compressed image frames the image information of which has not been determined using motion-compensated temporal prediction. Such frames are called INTRA-frames, or I-frames.

Correspondingly, motion-compensated video sequence image frames predicted from previous image frames, are called INTER-frames, or P-frames (Predicted). The image information of P-frames is determined using one I-frame and possibly one or more previously coded P-frames. If a frame is lost, frames dependent on it can no longer be correctly decoded.

[0008] An I-frame typically initiates a video sequence defined as a Group of Pictures (GOP), the P-frames of which can only be determined on the basis of the I-frame and the previous P-frames of the GOP in question. The next I-frame begins a new group of pictures GOP, the image information comprised by which cannot thus be determined on the basis of the frames of the previous GOP. In other words, groups of pictures are not temporally overlapping, and each group of picture can be decoded separately. In addition, many video compression methods employ bi-directionally predicted B-frames (Bi-directional), which are set between two anchor frames (I- and P-frames, or two P-frames) within a group of pictures GOP, the image information of a B-frame being predicted from both the previous anchor frame and the one succeeding the B-frame. B-frames therefore provide image information of higher quality than P-frames, but typically they are not used as anchor frames, and therefore their removal from the video sequence does not degrade the quality of subsequent images. However, nothing prevents B-frames from being used as anchor frames as well, only in that case they cannot be removed from the video sequence without deteriorating the quality of the frames dependent on them.

[0009] Each video frame may be divided into what are known as macroblocks that comprise the colour components (such as Y, U, V) of all pixels of a rectangular image area. More specifically, a macroblock consists of at least one block per colour component, the blocks each comprising colour values (such as Y, U or V) of one colour level in the image area concerned. The spatial resolution of the blocks may differ from that of the macroblocks, for example U- and V-components may be displayed using only half of the resolution of Y-component. Macroblocks can be further grouped into slices, for example, which are groups of macroblocks that are typically selected in the scanning order of the image. Temporal prediction is typically carried out in video coding methods block- or macroblock-specifically, instead of image-frame-specifically.

[0010] To allow for flexible streaming of video files, many video coding systems employ scalable coding in which some elements or element groups of a video sequence can be removed without affecting the reconstruction of other parts of the video sequence. Scalability is typically implemented by grouping the image frames into a number of hierarchical layers. The image frames coded into the image frames of the base layer substantially comprise only the ones that are compulsory for the decoding of the video information at the receiving end. The base layer of each group of pictures GOP thus comprises one I-frame and a necessary number of P-frames. One or more enhancement layers can be determined below the base layer, each one of the layers improving the quality of the video coding in comparison with an upper layer. The enhancement layers thus comprise P- or B-frames predicted on the basis of motion-compensation from one or more upper layer images. The frames are typically numbered according to an arithmetical series.

[0011] In streaming, transmission bit rate must be controllable either on the basis of the bandwidth to be used or the maximum decoding or bit rate value of the recipient. Bit rate can be controlled either at the streaming server or in some element of the telecommunications network, such as an Internet router or a base station of a mobile communications network. The simplest means for the streaming server to control the bit rate is to leave out B-frames having a high information content from the transmission. Further, the streaming server may determine the number of scalability layers to be transmitted in a video stream, and thus the number of the scalability layers can be changed always when a new group of pictures GOP begins. It is also possible to use different video sequence coding methods. Correspondingly, B-frames, as well as other P-frames of the enhancement layers, can be removed from the bit stream in a telecommunications network element.

[0012] The above arrangement involves a number of drawbacks. Many coding methods, such as the coding according to the ITU-T (International Telecommunications Union, Telecommunications Standardization Sector) standard H.263, are familiar with a procedure called reference picture selection. In reference picture selection at least a part of a P-image has been predicted from at least one other image than the one immediately preceding the P-image in the time domain. The selected reference image is signalled in a coded bit stream or in bit stream header fields image-,

image-segment- (such as a slice or a group of macroblocks), macroblock-, or block-specifically. The reference picture selection can be generalized such that the prediction can also be made from images temporally succeeding the image to be coded. Further, the reference picture selection can be generalized to cover all temporally predicted frame types, including B-frames. Since it is possible to also select at least one image preceding an I-image that begins a group of pictures GOP as the reference image, a group of pictures employing reference picture selection cannot necessarily be decoded independently. In addition, the adjusting of scalability or coding method in the streaming server or a network element becomes difficult, because the video sequence must be decoded, parsed and buffered for a long period of time to allow any dependencies between different image groups to be detected.

[0013] A further problem relates to detection of image frames from which a decoder can start the decoding process. The detection is useful for multiple purposes. For example, an end-user may wish to start browsing a video file from the middle of a video sequence. Another example relates to starting the reception of a broadcast or multicast video transmission from the middle of the video transmission. A third example relates to on-demand streaming from a server and occurs when an end-user wishes to start playback from a certain position of a stream.

BRIEF DESCRIPTION OF THE INVENTION

[0014] Now there is invented an improved method and equipment implementing the method, that enable detection of image frames from which a decoder can start the decoding process. Various aspects of the invention include methods, a video encoder, a video decoder, and computer programs that are characterized by what is stated in the independent claims. The preferred embodiments of the invention are disclosed in the dependent claims.

[0015] The invention is based on the idea of encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference picture is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order. An indication of at least one image frame is encoded into the video sequence, which indicated image frame is the first picture, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence. Respectively, in the decoding phase, the indication

of at least one image frame is decoded from the video sequence, and the decoding of the video sequence is started from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame.

[0016] Consequently, the idea of the present invention is to determine an initiation picture in an independently decodable group of pictures, whereby in the decoding phase any picture prior to said initiation picture is defined as unusable for reference. Accordingly, after decoding the initiation picture, all following pictures of the independently decodable sequence can be decoded without prediction from any picture decoded prior to said initiation picture.

[0017] According to an embodiment, the indication is encoded into the video sequence as a separate flag included in the header of a slice.

[0018] According to an embodiment, identifier values for pictures are encoded according to a numbering scheme, and the identifier value for the indicated first picture of an independent sequence is reset, preferably to zero.

[0019] According to an embodiment, an identifier value for the independent sequence is encoded into the video sequence.

[0020] An advantage of the procedure of the invention is that it enables to start browsing a video sequence from a random point, i.e. it provides the decoder with the information about the first picture of an independently decodable sequence. Thus, the decoder knows that by decoding this first picture, it is possible to continue the decoding process without any prediction from any prior picture. Accordingly, a further advantage is that the decoder may discard any picture decoded prior to said initiation picture from its buffer memory, since the prior pictures are no longer needed in the decoding process. A further advantage is that the procedure of the invention allows a separate video sequence to the easily inserted into another video sequence.

[0021] A further advantage is that enables identification of a picture boundary between two back-to-back initiation pictures by looking at the sub-sequence number of the initiation pictures. A yet further advantage relates to detection of losses of image frames that start an independently decodable sub-sequence. If such an image frame is lost e.g. during transmission, it is unlikely that any error concealment method would result into a subjectively satisfactory image quality. It is therefore an advantage that decoders are provided means

to detect losses of image frames that start an independently decodable sub-sequence. Decoders may react to such a loss by requesting a retransmission or picture refresh, for example.

BRIEF DESCRIPTION OF THE DRAWINGS

[0022] In the following, the invention will be described in connection with the preferred embodiments and with reference to the accompanying drawings, in which

Figure 1 illustrates a common multimedia data streaming system in which the scalable coding hierarchy of the invention can be applied;

Figure 2 illustrates a scalable coding hierarchy of a preferred embodiment of the invention;

Figures 3a and 3b illustrate embodiments of the invention for adjusting scalability;

Figures 4a, 4b and 4c illustrate embodiments of the invention for adjusting image numbering;

Figures 5a, 5b and 5c illustrate embodiments of the invention for using B-frames in a scalable coding hierarchy;

Figures 6a, 6b and 6c illustrate scalable coding hierarchies of preferred embodiments of the invention in connection with reference picture selection; and

Figure 7 illustrates an arrangement according to a preferred embodiment of the invention for coding scene transition.

DETAILED DESCRIPTION OF THE INVENTION

[0023] In the following, a general-purpose multimedia data streaming system is disclosed, the basic principles of which can be applied in connection with any telecommunications system. Although the invention is described here with a particular reference to a streaming system, in which the multimedia data is transmitted most preferably through a telecommunications network employing a packet-switched data protocol, such as an IP network, the invention can equally well be implemented in circuit-switched networks, such as fixed telephone networks PSTN/ISDN (Public Switched Telephone Network/Integrated Services Digital Network) or in mobile communications networks PLMN (Public Land Mobile Network). Further, the invention can be applied in the streaming of multimedia files in the form of both normal

streaming and progressive downloading, and for implementing video calls, for example.

[0024] It is also to be noted that although the invention is described here with a particular reference to streaming systems and the invention can also be advantageously applied in them, the invention is not restricted to streaming systems alone, but can be applied in any video reproduction system, irrespective of how a video file that is to be decoded is downloaded and where it is downloaded from. The invention can therefore be applied for example in the playback of a video file to be downloaded from a DVD disc or from some other computer memory carrier, for example in connection with varying processing capacity available for video playback. In particular, the invention can be applied to different video codings of low bit rate that are typically used in telecommunications systems subject to bandwidth restrictions. One example is the system defined in the ITU-T standard H.263 and the one that is being defined in H.26L (possibly later to become H.264). In connection with these, the invention can be applied to mobile stations, for example, in which case the video playback can be made to adjust both to changing transfer capacity or channel quality and to the processor power currently available, when the mobile station is used also for executing other applications than video playback.

[0025] It is further to be noted that, for the sake of clarity, the invention will be described below by giving an account of image frame coding and temporal predicting on image frame level. However, in practice coding and temporal predicting typically take place on block or macroblock level, as described above.

[0026] With reference to Figure 1, a typical multimedia streaming system will be described, which is a preferred system for applying the procedure of the invention.

[0027] A multimedia data streaming system typically comprises one or more multimedia sources 100, such as a video camera and a microphone, or video image or computer graphic files stored in a memory carrier. Raw data obtained from the different multimedia sources 100 is combined into a multimedia file in an encoder 102, which can also be referred to as an editing unit. The raw data arriving from the one or more multimedia sources 100 is first captured using capturing means 104 included in the encoder 102, which capturing means can be typically implemented as different interface cards,

driver software, or application software controlling the function of a card. For example, video data may be captured using a video capture card and the associated software. The output of the capturing means 104 is typically either an uncompressed or slightly compressed data flow, for example uncompressed video frames of the YUV 4:2:0 format or motion-JPEG image format, when a video capture card is concerned.

[0028] An editor 106 links different media flows together to synchronize video and audio flows to be reproduced simultaneously as desired. The editor 106 may also edit each media flow, such as a video flow, by halving the frame rate or by reducing spatial resolution, for example. The separate, although synchronized, media flows are compressed in a compressor 108, where each media flow is separately compressed using a compressor suitable for the media flow. For example, video frames of the YUV 4:2:0 format may be compressed using the low bit rate video coding according to the ITU-T recommendation H.263 or H.26L. The separate, synchronized and compressed media flows are typically interleaved in a multiplexer 110, the output obtained from the encoder 102 being a single, uniform bit flow that comprises data of a plural number of media flows and that may be referred to as a multimedia file. It is to be noted that the forming of a multimedia file does not necessarily require the multiplexing of a plural number of media flows into a single file, but the streaming server may interleave the media flows just before transmitting them.

[0029] The multimedia files are transferred to a streaming server 112, which is thus capable of carrying out the streaming either as real-time streaming or in the form of progressive downloading. In progressive downloading the multimedia files are first stored in the memory of the server 112 from where they may be retrieved for transmission as need arises. In real-time streaming the editor 102 transmits a continuous media flow of multimedia files to the streaming server 112, and the server 112 forwards the flow directly to a client 114. As a further option, real-time streaming may also be carried out such that the multimedia files are stored in a storage that is accessible from the server 112, from where real-time streaming can be driven and a continuous media flow of multimedia files is started as need arises. In such case, the editor 102 does not necessarily control the streaming by any means. The streaming server 112 carries out traffic shaping of the multimedia data as regards the bandwidth available or the maximum decoding and playback rate

of the client 114, the streaming server being able to adjust the bit rate of the media flow for example by leaving out B-frames from the transmission or by adjusting the number of the scalability layers. Further, the streaming server 112 may modify the header fields of a multiplexed media flow to reduce their size and encapsulate the multimedia data into data packets that are suitable for transmission in the telecommunications network employed. The client 114 may typically adjust, at least to some extent, the operation of the server 112 by using a suitable control protocol. The client 114 is capable of controlling the server 112 at least in such a way that a desired multimedia file can be selected for transmission to the client, in addition to which the client is typically capable of stopping and interrupting the transmission of a multimedia file.

[0030] When the client 114 is receiving a multimedia file, the file is first supplied to a demultiplexer 116, which separates the media flows comprised by the multimedia file. The separate, compressed media flows are then supplied to a decompressor 118 where each separate media flow is decompressed by a decompressor suitable for each particular media flow. The decompressed and reconstructed media flows are supplied to a playback unit 120 where the media flows are rendered at a correct pace according to their synchronization data and supplied to presentation means 124. The actual presentation means 124 may comprise for example a computer or mobile station display, and loudspeaker means. The client 114 also typically comprises a control unit 122 that the end user can typically control through a user interface and that controls both the operation of the server, through the above-described control protocol, and the operation of the playback unit 120, on the basis of the instructions given by the end user.

[0031] It is to be noted that the transfer of multimedia files from the streaming server 112 to the client 114 takes place through a telecommunications network, the transfer path typically comprising a plural number of telecommunications network elements. It is therefore possible that there is at least some network element that can carry out traffic shaping of multimedia data with regard to the available bandwidth or the maximum decoding and playback rate of the client 114 at least partly in the same way as described above in connection with the streaming server.

[0032] Scalable coding will be described below with reference to a preferred embodiment of the invention and an example illustrated in Figure 2. Figure 2 shows part of a compressed video sequence having a first frame 200,

which is an INTRA frame, or I-frame, and thus an independently determined video frame the image information of which is determined without using motion-compensated temporal prediction. The I-frame 200 is placed on a first scalability layer, which may be referred to as an INTRA layer. Each scalability layer is assigned a unique identifier, such as the layer number. The INTRA layer may therefore be given the number 0, for example, or some other alphanumeric identifier, for example a letter, or a combination of a letter and a number.

[0033] Correspondingly, sub-sequences consisting of groups of one or more video frames are determined for each scalability layer, at least one of the images in a group (typically the first or the last one) being temporally predicted at least from a video frame of another, sub-sequence of typically either a higher or the same scalability layer, the rest of the video frames being temporally predicted either from only the video frames of the same sub-sequence, or possibly also from one or more video frames of said second sub-sequence. A sub-sequence may de decoded independently, irrespective of other sub-sequences, except said second sub-sequence. The sub-sequences of each scalability layer are assigned a unique identifier using for example consecutive numbering starting with number 0 given for the first sub-sequence of a scalability layer. Since the I-frame 200 is determined independently and can also be decoded independently upon reception, irrespective of other image frames, it also forms in a way a separate sub-sequence.

[0034] An essential aspect of the present invention is therefore to determine each sub-sequence in terms of those sub-sequences the sub-sequence is dependent on. In other words, a sub-sequence comprises information about all the sub-sequences that have been directly used for predicting the image frames of the sub-sequence in question. This information is signalled in a video sequence bit stream, preferably separate from the actual image information, and therefore the image data of the video sequence can be preferably adjusted because it is easy to determine video sequence portions that are to be independently decoded and can be removed without affecting the decoding of the rest of the image data.

[0035] Next, within each sub-sequence, the video frames of the sub-sequence are given image numbers, using for example consecutive numbering that starts with the number 0 given to the first video frame of the sub-sequence. Since the I-frame 200 also forms a separate sub-sequence, its

image number is 0. In Figure 2, the I-frame 200 shows the type (I), sub-sequence identifier and image number (0.0) of the frame.

[0036] Figure 2 further shows a next I-frame 202 of the INTRA layer, the frame thus being also an independently determined video frame that has been determined without using motion-compensated temporal prediction. The temporal transmission frequency of I-frames depends on many factors relating to video coding, image information contents and the bandwidth to be used, and, depending on the application or application environment, I-frames are transmitted in a video sequence at intervals of 0.5 to 10 seconds, for example. Since the I-frame 202 can be independently decoded, it also forms a separate sub-sequence. Since this is the second sub-sequence in the INTRA-layer, the consecutive numbering of the sub-sequence identifier of the I-frame 202 is 1. Further, since the I-frame 202 also forms a separate sub-sequence, i.e. it is the only video frame in the sub-sequence, its image number is 0. The I-frame 202 can thus be designated with identifier (I.1.0.) Correspondingly, the identifier of the next I-frame on the INTRA layer is (I.2.0.), etc. As a result, only independently determined I-frames in which the image information is not determined using motion-compensated temporal prediction are coded into the first scalability layer, i.e. the INTRA layer. The sub-sequences can also be determined using other kind of numbering or other identifiers, provided that the sub-sequences can be distinguished from one another.

[0037] The next scalability layer, which has a layer number 1, for example, and which may be referred to as the base layer, comprises coded, motion-compensated INTER or P-frames typically predicted only from previous image frames, i.e. in this case from the I-frames of an upper INTRA layer. The image information of the first P-frame 204 of the base layer shown in Figure 2 is determined using the I-frame 200 of the INTRA layer. A P-frame 204 begins the first sub-sequence of the base layer, and therefore the sub-sequence identifier of the P-frame 204 is 0. Further, since the P-frame 204 is the first image frame of the first sub-sequence of the base layer, the image number of the P-frame 204 is 0. The P-frame 204 can thus be identified with (P.0.0).

[0038] The temporally succeeding P-frame 206 of the base layer is predicted from the previous P-frame 204. The P-frames 204 and 206 thus belong to the same sub-sequence, whereby the P-frame 206 also receives the sub-sequence identifier 0. Since the P-frame 206 is the second image frame in

the sub-sequence 0, the image number of the P-frame 206 is 1, and the P-frame 206 can be identified with (P.0.1).

[0039] The scalability layer succeeding the base layer and having the layer number 2, is called enhancement layer 1. This layer comprises coded, motion-compensated P-frames predicted only from previous image frames, in this case either from I-frames of the INTRA layer or P-frames of the base layer. Figure 2 shows a first image frame 208 and a second image frame 210 of enhancement layer 1, which are both predicted only from the first image frame 200 of the INTRA layer. The P-frame 208 begins the first sub-sequence of enhancement layer 1, the sub-sequence identifier of the P-frame thus being 0. Further, since the P-frame 208 is the first and only image frame in said sub-sequence, the P-frame 208 receives the image number 0. The P-frame 208 can thus be identified with (P.0.0.).

[0040] Since the second image frame 210 is also predicted only from the first image frame 200 of the INTRA layer, the P-frame 210 begins the second sub-sequence of enhancement layer 1 and the sub-sequence identifier of the P-frame 210 is therefore 1. Since the P-frame 210 is the first image frame in the sub-sequence, the image number of the P-frame 210 is 0. The P-frame can thus be identified with (P.1.0.). The temporally succeeding P-frame 212 of enhancement layer 1 is predicted from the previous P-frame 210. The P-frames 210 and 212 thus belong to the same sub-sequence, and therefore the P-frame also receives the sub-sequence identifier 1. The P-frame 212 is the second image frame in the sub-sequence 1, and therefore the P-frame receives the image number 1 and can be identified with (P.1.1).

[0041] The temporally fourth image frame 214 of enhancement layer 1 is predicted from the first image frame 204 of the base layer. The P-frame 214 thus begins a third sub-sequence of enhancement layer 1 and therefore the P-frame 214 receives the sub-sequence identifier 2. Further, since the P-frame 214 is the first and only image frame in the sub-sequence, the image number of the P-frame 214 is 0. The P-frame 208 can therefore be identified with (P.2.0).

[0042] Also the temporally fifth image frame 216 of enhancement layer 1 is predicted only from the first image frame 204 of the base layer, the P-frame 216 thus beginning the fourth sub-sequence of enhancement layer 1, and the sub-sequence identifier of the P-frame 216 is 3. In addition, since the P-frame 216 is the first one in the sub-sequence in question, the image

number of the P-frame 216 is 0. The P-frame 216 can therefore be identified with (P.3.0). The temporally following P-frame 218 of enhancement layer 1 is predicted from the previous P-frame 216. The P-frames 216 and 218 thus belong to the same sub-sequence, and the sub-sequence identifier of the P-frame 218 is also 3. Since the P-frame 218 is the second image frame in the sub-sequence 3, the image number of the P-frame 218 is 1 and the identifier of the P-frame 218 is (P.3.1).

[0043] For simplicity and clarity of illustration the above disclosure only relates to I- and P-frames. However, a person skilled in the art will find it apparent that the scalable video coding of the invention can also be implemented using other known image frame types, such as the above described B-frames and at least SI-frames, SP-frames and MH-frames. SI-frames correspond to I-frames, but together with an SP-frame they allow an identical image to be reconstructed. An SP-frame, in turn, is a P-frame subjected to a particular coding that allows an identical image to be reconstructed together with an SI-frame or another SP-frame. SP-frames are typically placed into a video sequence into points where an access point or a scanning point is desired, or where the changing of the coding parameters of the video stream should be possible. The frames can also be used for error correction and for increasing error tolerance. SP-frames are otherwise similar to ordinary P-frames predicted from previous frames, except that they are defined so that they can be replaced by another video frame of the SP- or SI-type, the result of the decoding of the new frame being identical with the decoding result of the original SP-frame that was in the video stream. In other words, a new SP-frame that is used for replacing the one that was in the video stream is predicted from another sequence or video stream and yet the reconstructed frame has identical contents. SP-frames are described for example in the Applicant's earlier application PCT/FI02/00004.

[0044] Similarly as B-frames, macroblocks of MH (Multi Hypothesis) frames, based on motion-compensated prediction, are predicted from two other frames, which are not, however, necessarily located next to an MH-frame. More precisely, the predicted macroblocks are computed as an average of two macroblocks of two other frames. Instead of two frames, MH-frame macroblocks can naturally be also predicted from one other frame. Reference images may change according to macroblock, in other words, all macroblocks

in one and the same image are not necessarily predicted using the same frames.

[0045] A sub-sequence thus covers a specific period of time in a video sequence. The sub-sequences of the same layer or of different layers may be partly or entirely overlapping. If there are temporally overlapping image frames on the same layer, the frames are interpreted as alternative presentations of the same image content and therefore any mode of image presentation can be used. On the other hand, if there are temporally overlapping image frames on different layers, they form different presentations of the same image content, and therefore presentations differ in image quality, i.e. the quality of image is better on a lower layer.

[0046] The above disclosure referring to Figure 2 illustrates a scalable coding arrangement and a hierarchical structure and numbering of image frames according to a preferred embodiment of the invention. In this embodiment the INTRA-layer only comprises I-frames and the base layer can only be decoded using the information received from the INTRA-layer. Correspondingly, the decoding of enhancement layer 1 typically requires information from both the base layer and the INTRA-layer.

[0047] The number of the scalability layers is not restricted to three, as above, but any number of enhancement layers that is considered necessary for producing sufficient scalability may be used. Consequently, the layer number of enhancement layer 2 is four, that of enhancement layer 3 is five, etc. Since some of the image frames in the above example are given the same identifier (e.g. the identifier of both image frames 204 and 208 is (P.0.0)), by including the layer number in the identifier each image frame can be uniquely identified and, at the same time, the dependencies of each image frame on other image frames is preferably determined. Each image frame is thus uniquely identified, the identifier of image frame 204, for example, being (P.1.0.0), or simply (1.0.0) and, correspondingly, that of image 208 being (P.2.0.0), or (2.0.0).

[0048] According to a preferred embodiment of the invention, the number of a reference image frame is determined according to a specific, pre-determined alpha-numeric series, for example as an integer between 0 and 255. When the parameter value achieves the maximum value N (e.g. 255) in the series concerned, the determining of the parameter value starts from the beginning, i.e. from the minimum value of the series (e.g. 0). An image frame is

thus uniquely identified within a specific sub-sequence up to the point where the same image number is used again. The sub-sequence identifier can also be determined according to a specific, predetermined arithmetic series. When the value of the sub-sequence identifier achieves the maximum value N of the series, the determining of the identifier starts again from the beginning of the series. However, a sub-sequence cannot be assigned an identifier that is still in use (within the same layer). The series in use may also be determined in another way than arithmetically. One alternative is to assign random sub-sequence identifiers, taking into account that an assigned identifier is not be used again.

[0049] A problem in the numbering of image frames arises when the user wishes to start browsing a video file in the middle of a video sequence. Such situations occur for example when the user wishes to browse a locally stored video file backward or forward or to browse a streaming file at a particular point; when the user initiates the playback of a streaming file from a random point; or when a video file that is to be reproduced is detected to contain an error that interrupts the playback or requires the playback to be resumed from a point following the error. When the browsing of a video file is resumed from a random point after previous browsing, discontinuity typically occurs in the image numbering. The decoder typically interprets this as unintentional loss of image frames and unnecessarily tries to reconstruct the image frames suspected as lost.

[0050] According to a preferred embodiment of the invention, this can be avoided in the decoder by defining an initiation image in an independently decodable Group of Pictures GOP that is activated at the random point of the video file, and the number of the initiation image is set at zero. This independently decodable image group can thus be a sub-sequence of an INTRA-layer, for example, in which case an I-frame is used as the initiation image, or, if scaling originating from the base layer is employed, the independently decodable image group is a sub-sequence of the base layer, in which case the first image frame of the sub-sequence, typically an I-frame, is usually used as the initiation image. Consequently, when activated at a random point, the decoder preferably sets the identifier of the first image frame, preferably an I-frame, of the independently decodable sub-sequence at zero. Since the sub-sequence to be decoded may also comprise other image frames whose identifier is zero (for example when the above described alpha-

numeric series starts from the beginning), the beginning of the sub-sequence, i.e. its first image frame, can be indicated to the decoder for example by a separate flag added to the header field of a slice of the image frame. This allows the decoder to interpret the image numbers correctly and to find the correct image frame that initiates the sub-sequence from the video sequence image frames.

[0051] The above numbering system provides only one example of how the unique image frame identification of the invention can be carried out so that interdependencies between the image frames are indicated at the same time. However, video coding methods in which the method of the invention can be applied, such as video coding methods according to the ITU-T standards H.263 and H.26L, employ code tables, which in turn use variable length codes. When variable length codes are used for coding layer numbers, for example, a lower code word index, i.e. a smaller layer number, signifies a shorter code word. In practice the scalable coding of the invention will be used in most cases in such a way that the base layer will consist significantly more image frames than the INTRA-layer. This justifies the use of a lower index, i.e. a smaller layer number, on the base layer than on the INTRA-layer, because the amount of coded video data is thereby advantageously reduced. Consequently, the INTRA-layer is preferably assigned layer number 1 and the base layer is given layer number 0. Alternatively, the code can be formed by using fewer bits for coding the base layer number than the INTRA-layer number, in which case the actual layer number value is not relevant in view of the length of the code created.

[0052] Further, according to a second preferred embodiment of the invention, when the number of the scalability layers is to be kept low, the first scalability layer in particular can be coded to comprise both the INTRA-layer and the base layer. From the point of view of coding hierarchy, the simplest way to conceive this is to leave out the INTRA-layer altogether, and to provide the base layer with coded frames consisting of both independently defined I-frames, the image information of which has not been determined using motion-compensated temporal prediction, and image frames predicted from previous frames, which image frames in this case are motion-compensated P-frames predicted from the I-frames of the same layer. The layer number 0 can thus still be used for the base layer and, if enhancement layers are coded into the video

sequence, enhancement layer 1 is assigned layer number 1. This is illustrated in the following, with reference to Figures 3a and 3b.

[0053] Figure 3a shows a non-scalable video sequence structure, in which all image frames are placed on the same scalability layer, i.e. the base layer. The video sequence comprises a first image frame 300 which is an I-frame (I.0.0) and which thus initiates a first sub-sequence. The image frame 300 is used for predicting a second image frame 302 of the sub-sequence, i.e. a P-frame (P.0.1), which is then used for predicting a third image frame 304 of the sub-sequence, i.e. a P-frame (P.0.2), which is in turn used for predicting the next image frame 306, i.e. a P-frame (P.0.3). The video sequence is then provided with an I-frame (I.1.0) coded therein, i.e. an I-frame 308, which thus initiates a second sub-sequence in the video sequence. This kind of non-scalable coding can be used for example when the application employed does not allow scalable coding to be used, or there is no need for it. In a circuit-switched videophone application, for example, channel bandwidth remains constant and the video sequence is coded in real-time, and therefore there is typically no need for scalable coding.

[0054] Figure 3b, in turn, illustrates an example of how scalability can be added, when necessary, to a combined INTRA- and base layer. Here, too, the video sequence base layer comprises a first image frame 310 which is an I-frame (I.0.0) and which initiates a first sub-sequence of the base layer. The image frame 310 is used for predicting a second image frame 312 of the sub-sequence, i.e. a P-frame (P.0.1), which is then used for predicting a third image frame 314 of the sub-sequence, i.e. a P-frame (P.0.2). Enhancement layer 1, however, is also coded into this video sequence and it comprises a first sub-sequence, the first and only image frame 316 of which is a P-frame (P.0.0), which is predicted from the first image frame 310 of the base layer. The first image frame 318 of a second sub-sequence of the enhancement layer is, in turn, predicted from the second image frame 312 of the base layer, and therefore the identifier of this P-frame is (P.1.0). The next image frame 320 of the enhancement layer is again predicted from the previous image frame 318 of the same layer and, therefore, it belongs to the same sub-sequence, its identifier thus being (P.1.1).

[0055] In this embodiment of the invention the sub-sequences of the base layer can be decoded independently, although a base layer sub-sequence may be dependent on another base layer sub-sequence. The

decoding of the base layer sub-sequences requires information from the base layer and/or from the second sub-sequence of enhancement layer 1, the decoding of the sub-sequences of enhancement layer 2 requires information from enhancement layer 1 and/or from the second sub-sequence of enhancement layer 2, etc. According to an embodiment, I-frames are not restricted to the base layer alone, but lower enhancement layers may also comprise I-frames.

[0056] The basic idea behind the above embodiments is that a sub-sequence comprises information about all the sub-sequences it is dependent on, i.e. about all sub-sequences that have been used for predicting at least one of the image frames of the sub-sequence in question. However, according to an embodiment it is also possible that a sub-sequence comprises information about all sub-sequences that are dependent on the sub-sequence in question, in other words, about all the sub-sequences in which at least one image frame has been predicted using at least one image frame of the sub-sequence in question. Since in the latter case the dependencies are typically determined temporally forward, image frame buffers can be advantageously utilized in the coding in a manner to be described later.

[0057] In all the above embodiments the numbering of the image frames is sub-sequence-specific, i.e. a new sub-sequence always starts the numbering from the beginning. The identification of an individual image frame thus requires the layer number, sub-sequence identifier and image frame number to be determined. According to a preferred embodiment of the invention, the image frames can be independently numbered using consecutive numbering in which successive reference image frames in the coding order are indicated with numbers incremented by one. As regards layer numbers and sub-sequence identifiers, the above-described numbering procedure can be used. This allows each image frame to be uniquely identified, when necessary, without using the layer number and sub-sequence identifier.

[0058] This is illustrated with the example shown in Figure 4a in which the base layer comprises a temporally first I-frame 400 (I.0.0). This frame is used for predicting a first image frame 402 of enhancement layer 1, i.e. (P.0.1), which is then used for predicting a second image frame 404 belonging to the same sub-sequence (with sub-sequence identifier 0), i.e. (P.0.2), which is used for predicting a third image frame 406 of the same sub-

sequence, i.e. (P.0.3), which is used for predicting a fourth image frame 408 (P.0.4) and, finally, the fourth frame for predicting a fifth image frame 410 (P.0.5). The temporally next video sequence image frame 412 is located on the base layer, where it is in the same sub-sequence as the I-frame 400, although temporally it is only the seventh coded image frame, and therefore its identifier is (P.0.6). The seventh frame is then used for predicting a first image frame 414 of the second sub-sequence of enhancement layer 1, i.e. (P.1.7), which is then used for predicting a second image frame 416 belonging to the same sub-sequence (with sub-sequence identifier 1), i.e. (P.1.8), which in tum used for predicting a third image frame 418 (P.1.9), the third for predicting a fourth image frame 420 (P.1.10) and, finally, the fourth for predicting a fifth image frame 422 (P.1.11) of the same sub-sequence. Again, the temporally next video sequence image frame 424 is located on the base layer, where it is in the same sub-sequence as the I-frame 400 and the P-frame 412, although temporally it is only the thirteenth coded image frame and therefore its identifier is (P.0.12). For clarity of illustration, the above description of the embodiment does not comprise layer identifiers, but it is apparent that in order to implement scalability, also the layer identifier must be signalled together with the video sequence, typically as part of the image frame identifiers.

[0059] Figures 4b and 4c show alternative embodiments for grouping the image frames of the video sequence shown in Figure 4a. The image frames in Figure 4b are numbered according to sub-sequence, i.e. a new sub-sequence always starts the numbering from the beginning (from zero). Figure 4c, in turn, employs image frame numbering which corresponds otherwise to that used in Figure 4a, except that the P-frames of the base layer are replaced by SP-frame pairs to allow for identical reconstruction of image information.

[0060] As stated above, the procedure of the invention can also be implemented using B-frames. One example of this is illustrated in Figures 5a, 5b and 5c. Figure 5a shows a video sequence in the time domain, the sequence comprising P-frames P1, P4 and P7, with B-frames placed between them, the interdependencies of the B-frames with regard to temporal predicting being shown with arrows. Figure 5b shows a preferred grouping of video sequence image frames in which the interdependencies shown in Figure 5a are indicated. Figure 5b illustrates sub-sequence-specific image frame numbering in which a new sub-sequence always starts the numbering of the

image frames from zero. Figure 5c, in turn, illustrates image frame numbering, which is consecutive in the order of temporal prediction, wherein the following reference frame always receives the next image number as the previously encoded reference frame. The image frame (B1.8) (and (B2.10)) does not serve as a reference prediction frame to any other frame, therefore it does not affect the image frame numbering.

[0061] The above examples illustrate different alternatives of how scalability of video sequence coding can be adjusted by using the method of the invention. From the point of view of the terminal device reproducing the video sequence, the more scalability layers are available, or the more scalability layers it is capable of decoding, the better the image quality. In other words, increase in the amount of image information and in the bit rate used for transferring the information improves the temporal or spatial resolution, or the spatial quality of the image data. Correspondingly, a higher number of scalability layers also sets considerably higher demands on the processing capacity of the terminal device performing decoding.

[0062] In addition, the above examples illustrate the advantage gained by using sub-sequences. With image frame identifiers, the dependencies of each image frame from other image frames in the sub-sequence are indicated in an unambiguous manner. A sub-sequence thus forms an independent whole that can be left out of a video sequence, when necessary, without affecting the decoding of subsequent image frames of the video sequence. In that case, only the image frames of the sub-sequence in question and of those sub-sequences on the same and/or lower scalability layers dependent on it are not decoded.

[0063] The image frame identifier data transmitted together with the video sequence are preferably included in the video sequence header fields or in the header fields of the transfer protocol to be used for transferring the video sequence. In other words, the identifier data of predicted image frames are not included in the image data of the coded video sequence, but always into the header fields, whereby the dependencies of the image frames can be detected without decoding the images of the actual video sequence. The identifier data of the image frames can be stored for example in the buffer memory of the streaming server as the video sequence is being coded for transmission. In addition, the sub-sequences can be independently decoded on each scalability

layer, because the image frames of a sub-sequence are not dependent on other sub-sequences of the same scalability layer.

[0064] According to an embodiment of the invention, the image frames comprised by a sub-sequence may thus depend also on other sub-sequences of the same scalability layer. This dependency must then be signalled for example to the streaming server carrying out traffic shaping, because interdependent sub-sequences located on the same layer cannot be separately removed from a video sequence to be transmitted. A preferred way to carry out the signalling is to include it in the image frame identifiers to be transmitted, for example by listing the layer-sub-sequence pairs the sub-sequence in question depends on. This also provides a preferred way of indicating dependency from another sub-sequence of the same scalability layer.

[0065] The above examples illustrate a situation where image frames are temporally predicted from previous image frames. In some coding methods, however, the reference picture selection has been further extended to also include the predicting of the image information of image frames from temporally succeeding image frames. Reference picture selection offers most diversified means for creating different temporally scalable image frame structures and allows the error sensitivity of the video sequence to be reduced. One of the coding techniques based on reference picture selection is INTRA-frame postponement. The INTRA-frame is not placed into its temporally "correct" position in the video sequence, but its position is temporally postponed. The video sequence image frames that are between the "correct" position of the INTRA-frame and its actual position are predicted temporally backward from the INTRA-frame in question. This naturally requires that uncoded image frames be buffered for a sufficiently long period of time so that all image frames that are to be displayed can be coded and arranged into their order of presentation. INTRA-frame transfer and the associated determining of sub-sequences in accordance with the invention are illustrated in the following with reference to Figure 6.

[0066] Figure 6a shows a video sequence part in which the INTRA-frame comprises a single I-frame 600, which is temporally transferred to the position shown in Figure 6, although the "correct" position of the I-frame in the video sequence would have been at the first image frame. The video sequence image frames between the "correct" position and the real position 600 are thus

temporally predicted backward from the I-frame 600. This is illustrated by a sub-sequence coded into enhancement layer 1 and having a first temporally backward predicted image frame 602, which is a P-frame (P.0.0). This frame is used for temporally predicting a previous image frame 604, i.e. a P-frame (P.0.1), which is used in turn for predicting an image frame 606, i.e. a P-frame (P.0.2), and, finally, the frame 606 for predicting an image frame 608, i.e. a P-frame (P.0.3), which is at the position that would have been the "correct" position of the I-frame 600 in the video sequence. Correspondingly, the I-frame 600 on the base layer is used for temporally forward prediction of a sub-sequence comprising four P-frames 610, 612, 614 and 616, i.e. P-frames (P.0.0), (P.0.1), (P.0.2) and (P.0.3).

[0067] The fact that in this example backward predicted image frames are placed on a lower layer than forward predicted image layers indicates that for purposes of illustration, backward predicted image frames are in this coding example considered subjectively less valuable than forward predicted image frames. Naturally the sub-sequences could both be placed on the same layer, in which case they would be considered equal, or a backward predicted sub-sequence could be on the upper layer, in which case it would be considered subjectively more valuable.

[0068] Figures 6b and 6c show some alternatives for coding a video sequence according to Figure 6a. In Figure 6b both forward and backward predicted sub-sequences are placed on the base layer, the I-frame being only located on the INTRA-layer. The forward predicted sub-sequence on this layer is thus the second sub-sequence and its sub-sequence identifier is 1. In Figure 6c, in turn, an I-frame and a forward predicted sub-sequence based on it are located on the base layer, while a backward predicted sub-sequence is located on enhancement layer 1.

[0069] Moreover, according to a preferred embodiment of the invention, the above-described scalability can be utilized for coding what is known as a scene transition into a video sequence. Video material, such as news reports, music videos and movie trailers, often comprise rapid cuts between separate image material scenes. Sometimes the cuts are abrupt, but often a procedure known as scene transition is used in which transfer from one scene to another takes place by dimming, wiping, mosaic dissolving or scrolling the image frames of a previous scene, and, correspondingly, by presenting those of a later scene. From the point of view of coding efficiency,

the video coding of a scene transition is often most problematic, because the image frames appearing during the scene transition comprise information on the image frames of both the terminating and the initiating scene.

[0070] A typical scene transition, fading, is carried out by gradually reducing the intensity or luminance of the image frames of a first scene to zero, while gradually increasing the intensity of the image frames of a second scene to its maximum value. This scene transition is referred to as cross-faded scene transition.

[0071] Generally speaking, a computer-made image can be thought of as consisting of layers, or image objects. Each object can be defined with reference to at least three information types: the structure of the image object, its shape and transparency, and the layering order (depth) in relation to the background of the image and to other image objects. Shape and transparency are often determined using what is known as an alpha plane, which measures opacity and the value of which is usually determined separately for each image object, possibly excluding the background, which is usually determined as non-transparent. The alpha plane value of a non-transparent image object, such as the background, can thus be set at 1.0, whereas the alpha plane value of a fully transparent image object is 0.0. The values in between define the intensity of the visibility of a specific image object in a picture in proportion to the background and to other, at least partly overlapping, image objects that have a higher depth value than the image object in question.

[0072] The superimposition of image objects in layers according to their shape, transparency and depth position is referred to as scene composition. In practice the procedure is based on the use of weighted averages. First, the image object that is closest to the background, i.e. deepest according to its depth position, is placed onto the background and a combined image is formed of the two. The pixel values of the combined image are formed as an average weighted by the alpha plane values of the background image and the image object in question. The alpha plane value of the combined image is then set at 1.0, after which it serves as a background image for the next image object. The process continues until all image objects are attached to the image.

[0073] In the following, a procedure according to a preferred embodiment of the invention will be described in which video sequence scalability layers are combined with the above described image objects of

image frames and their information types to provide a scene transition with scalable video coding that also has good compression efficiency.

[0074] This embodiment of the invention is illustrated in the following by way of example and in a simplified manner by using cross-faded scene transition, on one hand, and abrupt scene transition, on the other hand, as examples. The image frames to be displayed during a scene transition are typically formed of two superimposed image frames, a first image frame comprising a first image scene and a second image frame a second scene. One of the image frames serves as the background image and other, which is referred to as a foreground image, is placed on top of the background image. The opacity of the background image, i.e. its non-transparency value, is constant. In other words, its pixel-specific alpha plane values are not adjusted.

[0075] In this embodiment of the invention, the background and foreground images are both defined according to scalability layer. This is illustrated in Figure 7, which shows an example of how image frames of two different scenes can be placed on scalability layers during a scene transition of the invention. Figure 7 shows a first image frame 700 of a first (terminating) scene positioned on the base layer. The image frame 700 may be either an I-frame containing image information that has not been determined using motion-compensated temporal predicting, or it may be a P-frame that is a motion-compensated image frame predicted from previous image frames. The coding of a second (initiating) scene starts during the temporally following image frame and, according to the invention, the image frames of the scene are also placed on the base layer. Remaining image frames 702, 704 of the second (terminating) scene are then placed on enhancement layer 1. These image frames are typically P-frames.

[0076] In this embodiment, the image frames of the second (initiating) scene are thus placed on the base layer, at least for the duration of the scene transition. The first image frame 706 of the scene is typically an I-frame, and it is used for temporally predicting the succeeding image frames of the scene. Consequently, the succeeding image frames of the second scene are temporally predicted frames, typically P-frames, such as frames 708 and 710 shown in Figure 7.

[0077] According to a preferred embodiment of the invention, this placing of image frames on scalability layers can be used for implementing a cross-faded scene transition by determining the image layer that is on the base

layer always as a background image of maximum opacity (100%), or non-transparency value. During a scene transition, image frames located on enhancement layers are placed onto the background image and their opacity is adjusted for example by means of suitable filters such that the frames gradually change from non-transparent to transparent.

[0078] In the video sequence of Figure 7, there are no image frames on the lower scalability layers during the first base layer image frame 700. For this time instant, the first image frame 700 is only coded into the video sequence.

[0079] The next image frame 706 of the base layer initiates a new (second) scene, during which the image frame 706 is provided with depth positioning that places it as the background image, and its opacity value is set to the maximum. Temporally simultaneously with the image frame 706 of the base layer, there is an image frame 702 of a terminating (first) scene on enhancement layer 1. To allow a cross-faded scene transition to be produced, the transparency of the frame 702 must be increased. The example of Figure 7 assumes that the opacity of the image frame 702 is set at 67% and, in addition, the image frame 702 is provided with depth positioning that determines it as a foreground image. For this time instant, an image combining the image frames 706 and 702 is coded into the video sequence, image 706 being visible as a weaker image on the background and image 702 as a stronger image at the front, because its opacity value is essentially high (67%).

[0080] During the temporally following image frame, there is a second image frame 708 of the second scene on the base layer, the frame 708 being thus correspondingly provided with depth positioning determining it as a background image, and its opacity value is set to the maximum. Enhancement layer 1 further comprises the last image frame 704 of a temporally simultaneously terminating (first) scene, the opacity value of the frame being set at 33% and, in addition, the image frame 704 being provided with depth positioning that determines it as a foreground image as well. Consequently, for this time instant, an image combined of the image frames 708 and 704 is coded into the video sequence, the image 708 being displayed as a stronger image on the background and the image 704 as a weaker image on the foreground, because the opacity value of the image 704 is no longer more than 33%.

[0081] During the temporally following image frame, the base layer comprises a third image frame 710 of the second scene. Since the first scene has terminated, only the image frame 710 is coded into the video sequence, and the displaying of the second scene continues from the frame 710.

[0082] The above disclosure describes, by way of example, the positioning of image frames according to the invention on scalability layers to implement cross-faded scene transition in a manner that is advantageous from the point of view of coding efficiency. However, it is possible that when a video sequence is being transmitted or decoded, a situation arises in which the bit rate of the video sequence must be adjusted according to the maximum value of the bandwidth and/or terminal device decoding rate available for data transfer. This kind of bit rate control causes problems when the scene transition is to be implemented using prior art video coding methods.

[0083] A preferred embodiment of the present invention now allows one or more scalability layers, or independently decodable sub-sequences included in them, to be removed from a video sequence, whereby the bit rate of the video sequence can be decreased and yet, at the same time, the video sequence can be decoded without reducing image frequency. In the image frame positioning according to Figure 7, this can be implemented by removing enhancement layer 1 from the video sequence. The video sequence is thus only used for displaying the image frames 700, 706, 708 and 710 of the base layer. In other words, a direct transition from the first (terminating) scene to the second (initiating) scene takes place in the form of an abrupt scene transition, i.e. directly from the image frame 700 of the first scene into the I-image frame 706 that initiates the second scene. The transition is thus not a cross-faded scene transition but an abrupt scene transition. Nevertheless, the scene transition can be carried out in an advantageous manner without affecting the quality of the video sequence image, and the viewer usually does not experience an abrupt scene transition carried out instead of a cross-faded scene transition in any way disturbing or faulty. On the contrary, since the prior art implementation does not allow scalability layers to be removed, scene transition would often require image frequency to be reduced, which the viewer would find jerky and disturbing.

[0084] The invention thus provides a preferred means for carrying out multimedia data traffic shaping in a streaming server comprising information about the different sub-sequences of a video sequence: their

average bit rate, location in relation to the entire video sequence, duration and their interdependencies regarding the layers. The streaming server also determines the maximum value of the bandwidth available for the data transfer and/or the decoding rate of the terminal device. On the basis of this information, the streaming server decides how many scalability layers and which sub-sequences are transmitted in the video sequence. Bit rate control can thus be carried out, when necessary, by making first a rough adjustment of the number of the scalability layers, after which finer sub-sequence-specific adjustment can be easily carried out. At its simplest, bit rate control means making sub-sequence-specific decisions on whether a particular sub-sequence will be added to a video sequence or removed from it. In case of removal it is advisable to remove entire sub-sequences from a video sequence, because the removal of separate images may cause errors in other images of the same sub-sequence. For the same reason, all sub-sequences of a lower enhancement layer should be left out if they are dependent on the removed sub-sequence of a higher layer. If there are interdependent sub-sequences on one and the same scalability layer, sub-sequences dependent on an earlier sub-sequence must be removed if the earlier sub-sequence is removed.

[0085] If the image frame identifier data are added to a video sequence that is to be transmitted, traffic shaping can also be carried out in a telecommunications network element to be used for the transfer of the video sequence, for example in an Internet router, in different gateways, or at a base station or base station controller of a mobile communications network. For the network element to be able to maintain and process the sub-sequence information, it must have extra memory and processing capacity. For this reason traffic shaping that is to be carried out in the network is perhaps most probably executed using simple processing methods, such as the DiffServ, i.e. differentiated services, procedure that is supported by some IP-based networks. In the DiffServ method, each IP data packet is assigned a priority, whereby data packets of a higher priority are relayed faster and more reliably to the recipient than packets of a lower priority. This is advantageously applied to the scalability of the invention by determining not only scalability-layer-specific, but also sub-sequence-specific priorities, which enables a highly advanced priorisation.

[0086] There are many alternatives for adding image frame identifier data to a video sequence that is to be transmitted. In addition, it is also

possible not to include any identifier data into the video sequence, in which case traffic shaping is only carried out at the streaming server. The identifier data can be included in the header fields of a video sequence, or in the header fields of the transfer protocol to be used, such as RTP (Real Time Protocol). According to a preferred embodiment, the identifier data can be transferred using a Supplemental Enhancement Information (SEI) mechanism. SEI provides a data delivery mechanism that is transferred synchronously with the video data content, thus assisting in the decoding and displaying of the video sequence. The SEI mechanism, particularly when used for transferring layer and sub-sequence information, is disclosed more in detail in the ITU-T standard document ITU-T Rec. H.264 (ISO/IEC 14496 – 10:2002), Annex D. In the cases, wherein a separate transfer protocol or mechanism is used for identifier data transfer, traffic shaping can be carried out also at one of the network elements of the transfer path. In addition, the receiving terminal device can control the decoding.

[0087] If the encoder or decoder supports reference picture selection, video sequence coding requires that decoded image frames be buffered before the coding so as to allow the relationships between different image frames to be temporally predicted from one or more other image frames. Image frame buffering can be arranged at least in two different ways, either as sliding windowing or as adaptive buffer memory control. In sliding windowing, M image frames that were coded last are used as a buffer. The frames in the buffer memory are in a decoded and reconstructed form, which allows them to be used as reference images in the coding. As the coding proceeds, the image frame buffering functions on the basis of the FIFO principle (First-In-First-Out). Images that are not used as reference images, such as conventional B-images, do not need to be stored in the buffer. Alternatively, the buffering can be also be implemented as adaptive buffer memory control, in which case the image buffering is not restricted to the FIFO principle, but image frames that are not needed can be removed from the buffer in the middle of the process, or, correspondingly, some image frames can be stored in the buffer for a longer period of time, if they are needed as reference images for later image frames. A known reference picture selection is implemented by indexing image frames that are in the buffer memory into a specific order, the image indices being then used to refer to an image in connection with motion-compensation, for example. This indexing method generally provides better compression

efficiency compared to using image numbers, for example, for referring to a specific image when motion-compensation reference images are to be signalled.

[0088] The above reference image indexing method is sensitive to transfer errors, because the buffers of the sender's encoder and the recipient's decoder must contain mutually corresponding reconstructed images in identical order to ensure that the encoder and decoder both form the same indexing order. If the image frames are indexed in different order in the buffers of the encoder and the decoder, an incorrect reference image may be used in the decoder. To prevent this, it is essential that the decoder can be controlled to take into account image frames and sub-sequences that the encoder has intentionally removed from the video sequence. In that case the image frame numbering may comprise gaps, which the decoder typically interprets as errors and tries to reconstruct the image frames interpreted as lost. For this reason, it is essential that the encoder is capable to inform the decoder that the discontinuities in the image numbering of the transmitted image frames are intentional.

[0089] In response to this, and provided that sliding windowing is used for buffering the image frames, the decoder enters into the buffer memory a number of image frames, the contents of which may be fully random, corresponding to the missing image numbers. These random image frames are then designated by an identifier "invalid" to indicate that the frames in question do not belong to the actual video sequence, but are only filler frames entered for purposes of buffer memory management. A filler frame can naturally be implemented using only memory indicators, i.e. no data is preferably entered into the buffer memory, but memory management is used merely to store a reference to a generic "invalid" frame. The entering of the image frames of the actual video sequence continues from the correct image frame number after the number of filler frames indicated by the missing image numbers has been entered into the buffer, which allows the buffer memories of the encoder and the decoder to be kept preferably in synchronism. If during decoding a reference to an image number is detected which is then found to indicate a filler frame located in the buffer, error correction actions are initiated in the decoder to reconstruct the actual reference image, for example by asking the encoder to re-transmit the reference image in question.

[0090] Further, the procedure of the invention allows separate buffer memories to be used on the different scalability layers, or, correspondingly, sub-sequence-specifically. Each scalability layer may thus have a separate buffer memory that is conceptually separate and functions on the basis of the sliding window principle. Similarly, each sub-sequence may also be provided with a conceptually separate buffer memory that also functions on the basis of the sliding window principle. This means that the buffer memory is always emptied when a sub-sequence terminates. Separate buffer memories can be used in a preferred manner for reducing the need for signalling in certain situations in which ordinary sliding window buffering would be inadequate and actively adaptive buffer memory management would need to be used instead.

[0091] The H.26L standard defines a picture order count as a picture position in output order. The decoding process specified in the H.26L standard uses picture order counts to determine default index orderings for reference pictures in B slices, to represent picture order differences between frames and fields for vector scaling in motion vector prediction and for implicit mode weighted prediction in B slices, and to determine when successive slices in decoding order belong to different pictures. The picture order count is coded and transmitted for each picture.

[0092] In one embodiment of the invention, the decoder uses the picture order count to conclude that pictures are temporally overlapping, i.e., pictures that have an equal picture order count are temporally overlapping. Preferably, the decoder outputs only the picture on the highest received layer. In the absence of layer information, the decoder concludes that the latest temporally overlapping picture in decoding order resides on highest received layer.

[0093] The above disclosure describes a procedure for coding video frames for the purpose of producing a scalable, compressed video sequence. The actual procedure is carried out in a video encoder, such as the compressor 108 of Figure 1, which may be any known video encoder. For example a video encoder according to the ITU-T recommendation H.263 or H.26L may be used, the video encoder being arranged to form, in accordance with the invention, a first sub-sequence into a video sequence, at least part of the sub-sequence being formed by coding I-frames; to form at least a second sub-sequence into the video sequence, at least part of the sub-sequence being formed by coding at least P- or B-frames, and at least one video frame of

the second sub-sequence being predicted from at least one video frame of the first sub-sequence; and to determine into the video sequence the identification data of at least the video frames of the second sub-sequence.

[0094] According to the procedure of the invention, each sub-sequence of a particular scalability layer is preferably independently decodable, naturally taking into account dependencies from higher scalability layers and possibly other sub-sequences of the same scalability layer. A scalably compressed video sequence such as the one described above can thus be decoded by decoding a first sub-sequence of a video sequence, at least part of the sub-sequence having been formed by coding at least I-frames, and by decoding at least a second sub-sequence of the video sequence, at least part of the second sub-sequence having been formed by coding at least P- or B-frames, and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence, and by determining the identification and dependency data of at least the video frames comprised by the second sub-sequence of the video sequence, and by reconstructing at least part of the video sequence on the basis of the sub-sequence dependencies.

[0095] The actual decoding takes places in the video decoder, such as the decompressor 118 of Figure 1, which may be any known video decoder. For example, a low bit rate video decoder according to the ITU-T recommendation H.263 or H.26L may be used, which in this invention is arranged to decode a first sub-sequence of a video sequence, at least part of the sub-sequence having been formed by coding I-frames; to decode at least a second sub-sequence of the video sequence, at least part of the second sub-sequence having been formed by coding at least P- or B-frames and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence. The video decoder is arranged to determine the identification and dependency data of at least the video frames comprised by the second sub-sequence of the video sequence and to reconstruct at least part of the video sequence on the basis of the dependencies of the sub-sequences.

[0096] An essential aspect in the operation of the streaming system of the invention is that the encoder and decoder are positioned at least so that the encoder is operationally connected to the streaming server and the decoder is operationally connected to the receiving terminal device. However,

the different elements of the streaming system, terminal devices in particular, may include functionalities that allow two-way transfer of multimedia files, i.e. transmission and reception. The encoder and decoder can thus be implemented in the form of what it known as a video codec integrating both encoder and decoder functionalities.

[0097] It is to be noted that according to the invention the functional elements of the above described streaming system and its elements, such as the streaming server, video encoder, video decoder and terminal are preferably implemented by means of software, by hardware solutions, or as a combination of the two. The coding and decoding methods of the invention are particularly suitable for implementation as computer software comprising computer-readable commands for executing the process steps of the invention. A preferred way of implementing the encoder and the decoder is to store them in a storage means as a program code that can be executed by a computer-like device, for example a personal computer (PC) or a mobile station, to provide coding/decoding functionalities on the device in question.

[0098] Another alternative is to implement the invention as a video signal comprising a scalably compressed video sequence which in turn comprises video frames coded according to at least a first and a second frame format, the video frames according to the first frame format being independent of other video frames, and the video frames of the second frame format being predicted from at least one of the other video frames. According to the invention, the video signal in question comprises at least a first sub-sequence, at least part of which has been formed by coding at least video frames of the first frame format; at least a second sub-sequence, at least part of which has been formed by coding at least video frames of the second frame format; and at least one video frame of the second sub-sequence having been predicted from at least one video frame of the first sub-sequence; and at least one data field that determines video frames belonging to the second sub-sequence.

[0099] It is apparent to a person skilled in the art that as technology advances the basic idea of the invention can be implemented in various ways. The invention and its embodiments are therefore not restricted to the above examples, but they may vary within the scope of the claims.

CLAIMS

1. A method for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order, the method comprising:

encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence.

2. A method according to claim 1, further comprising:

encoding the indication into the video sequence as a separate flag included in the header of a slice.

3. A method according to claim 1, further comprising:

encoding identifier values for image frames according to a numbering scheme, and

resetting the identifier value for the indicated first image frame of an independent sequence.

4. A method according to claim 1, further comprising:

encoding into the video sequence an identifier value for the independent sequence.

5. A video encoder for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order, the video encoder being arranged to

encode into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence.

6. A video encoder according to claim 5, wherein the video encoder is arranged to

encode the indication into the video sequence as a separate flag included in the header of a slice.

7. A video encoder according to claim 5, wherein the video encoder is arranged to

encode identifier values for image frames according to a numbering scheme, and

reset the identifier value for the indicated first image frame of an independent sequence.

8. A video encoder according to claim 5, wherein the video encoder is arranged to

encode into the video sequence an identifier value for the independent sequence.

9. A computer program product, stored on a computer readable medium and executable in a data processing device, for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order, the computer program product comprising:

a computer program code for encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence.

10. A method for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order, the method comprising

decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame; and

starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame.

11. A method according to claim 10, wherein the indication is a separate flag included in the header of a slice.

12. A method according to claim 10, further comprising:

decoding identifier values for image frames according to a numbering scheme, and

resetting the identifier value for the indicated first image frame of an independent sequence.

13. A video decoder for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order, the video decoder being arranged to

decode from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame; and

start the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame.

14. A computer program product, stored on a computer readable medium and executable in a data processing device, for decoding a compressed video sequence, wherein at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order, the computer program product comprising:

a computer program code for decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, said sequence including the at least one reference image frame; and

a computer program code for starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame.

15. A video signal comprising a compressed video sequence including an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order, the video signal further comprising:

an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence.

16. A video signal according to claim 15, the video signal further comprising an identifier value for the independent sequence.

ABSTRACT

A method for encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference image frame is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order. An indication of at least one image frame is encoded into the video sequence, which indicated image frame is the first image frame, in decoding order, of the independent sequence, said at least one reference image frame being included in the sequence. In the decoding phase, the indication of at least one image frame is decoded from the video sequence, and the decoding of the video sequence is started from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame.

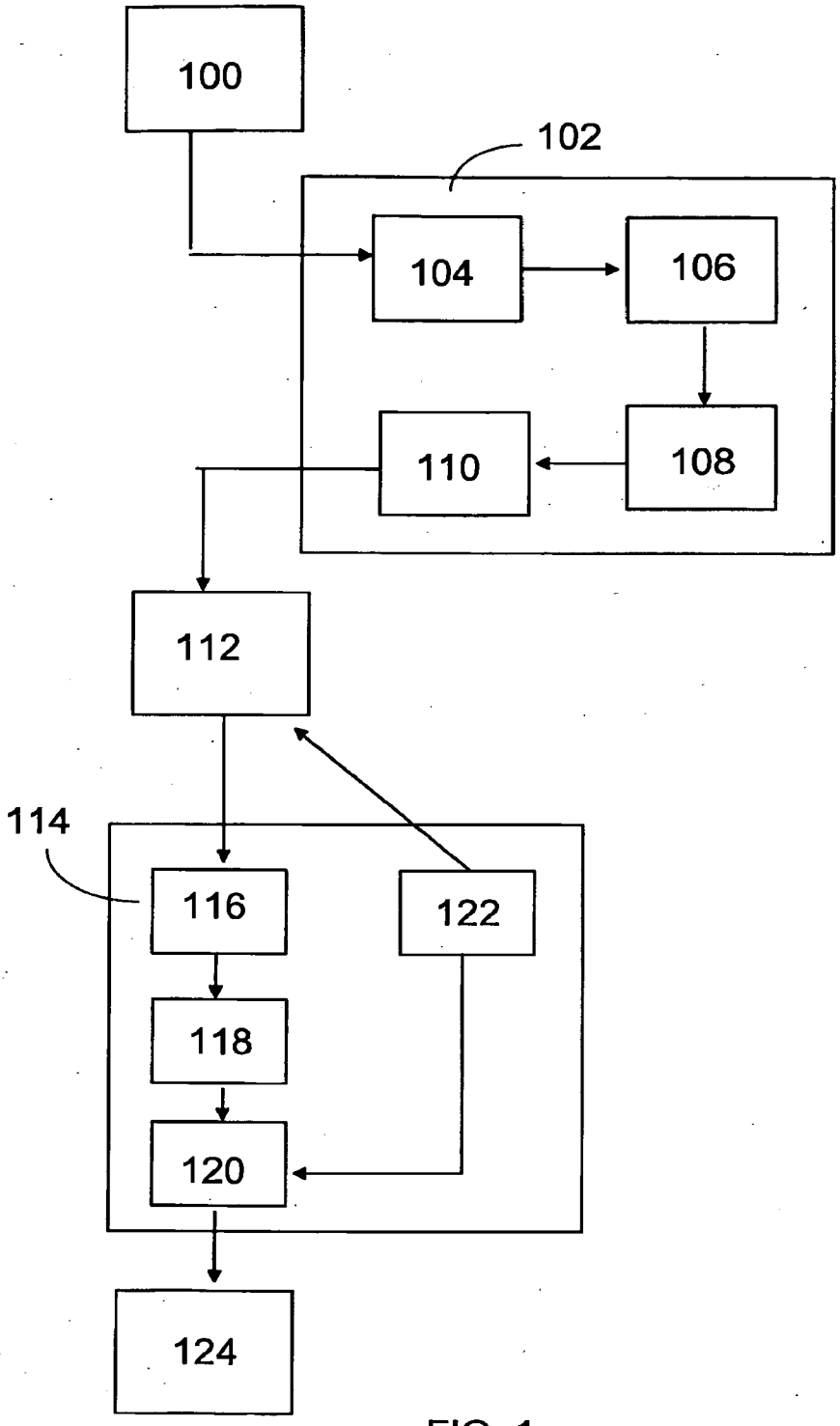


FIG. 1

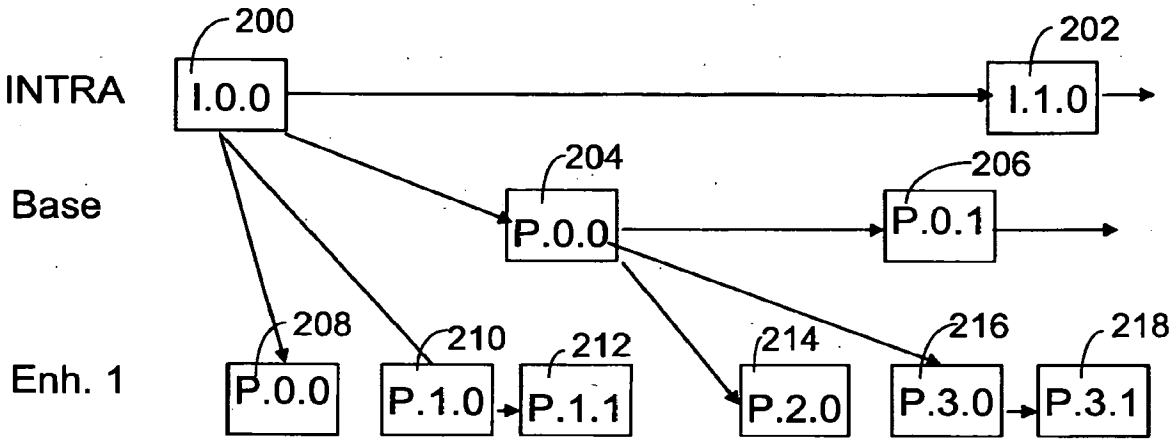


FIG. 2

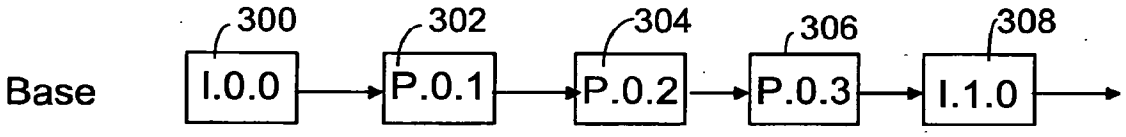


FIG. 3a

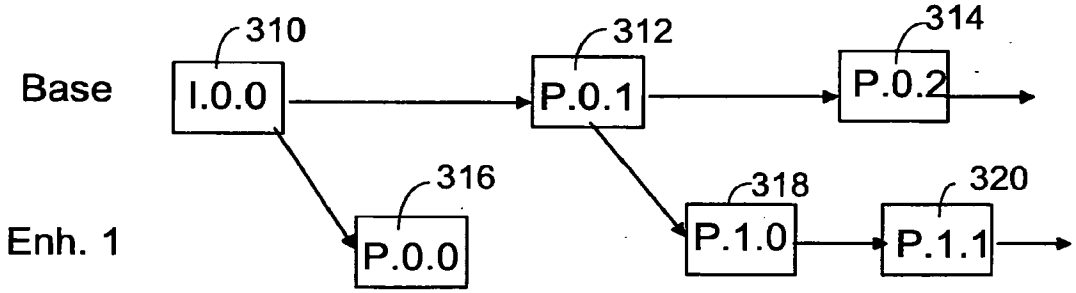


FIG. 3b

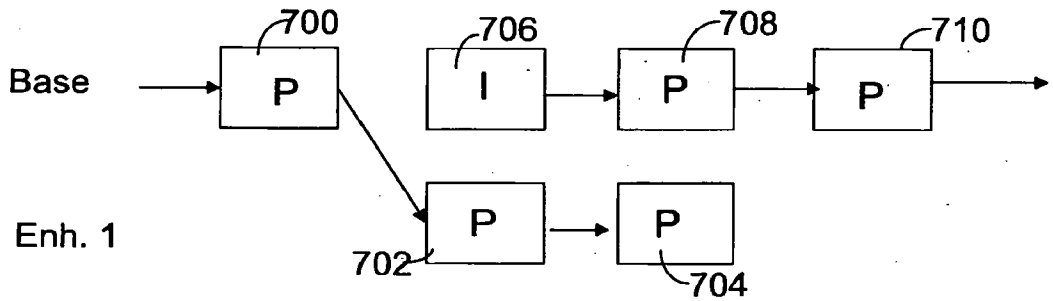


FIG. 7

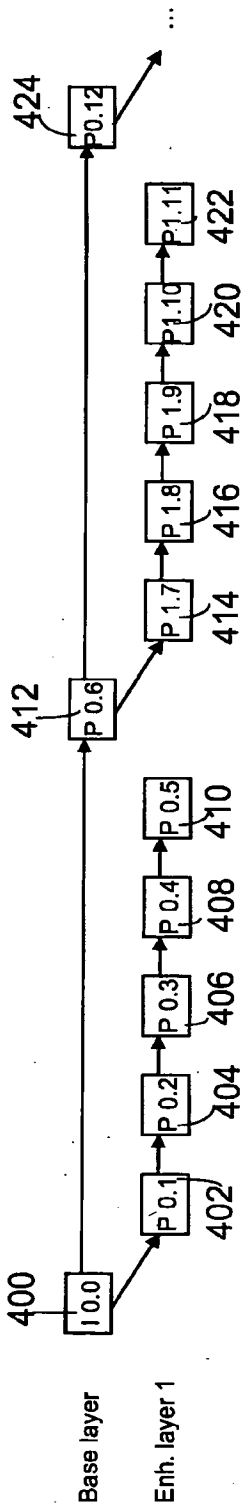


FIG. 4a

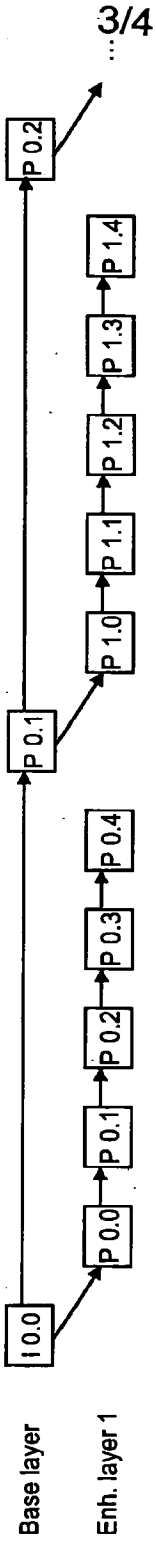


FIG. 4b

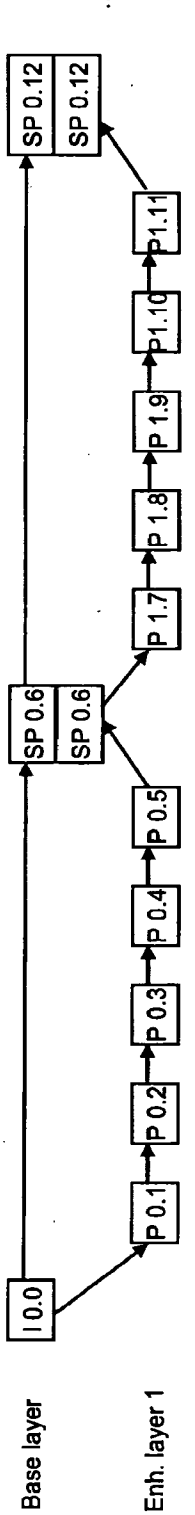


FIG. 4c

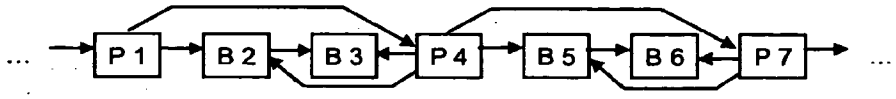


FIG. 5a

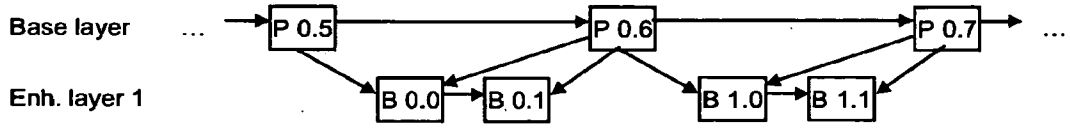


FIG. 5b

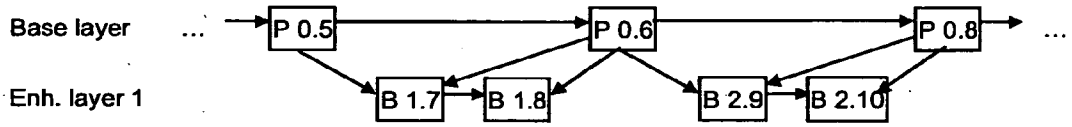


FIG. 5c

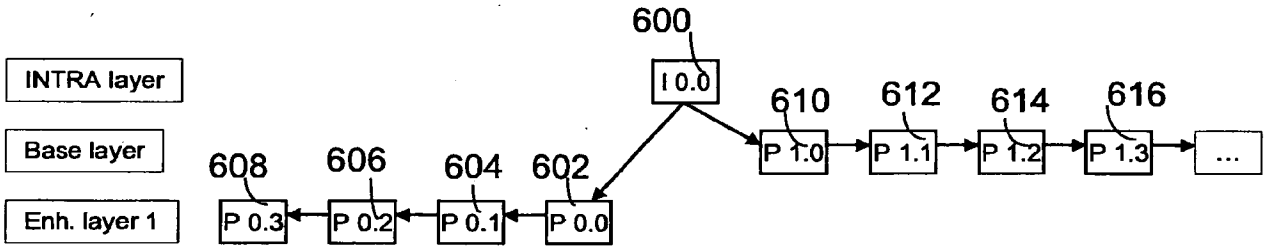


FIG. 6a

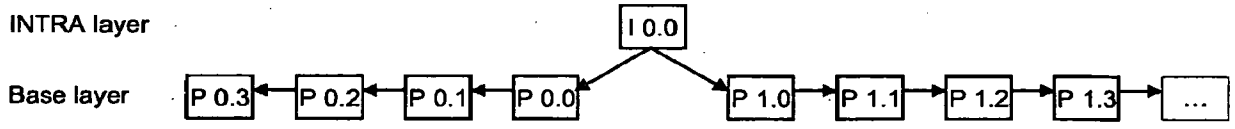


FIG. 6b

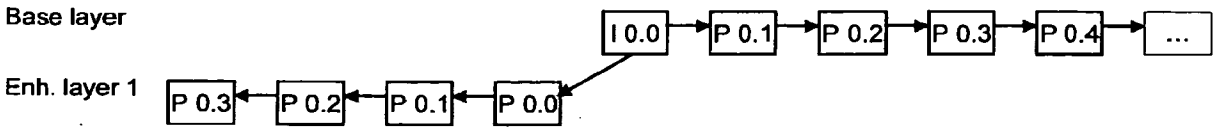


FIG. 6c

CRAWFORD MAUNU PLLC

United States Patent Application

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below named inventor I hereby declare that: my residence, post office address and citizenship are as stated below next to my name; that

I verily believe I am the original; first and sole inventor (if only one name is listed below) or a joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: **GROUPING OF IMAGE FRAMES IN VIDEO CODING.**

The specification of which
a. ☐ is attached hereto
b. ☒ is entitled **GROUPING OF IMAGE FRAMES IN VIDEO CODING,** having attorney docket number {**KOLS.021PA (NC 35273US)**}.
c. ☒ was filed on November 29, 2002 as application serial no. 10/306,942 and was amended on (if applicable) (in the case of a PCT-filed application) described and claimed in international no. filed and as amended on (if any), which I have reviewed and for which I solicit a United States patent.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, § 1.56 (attached hereto).

I hereby claim foreign priority benefits under Title 35, United States Code, § 119/365 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on the basis of which priority is claimed:

a. ☐ no such applications have been filed.
b. ☒ such applications have been filed as follows:

FOREIGN APPLICATION(S), IF ANY, CLAIMING PRIORITY UNDER 35 USC § 119			
COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	DATE OF ISSUE (day, month, year)
FINLAND	20020127	23 JANUARY 2002	
ALL FOREIGN APPLICATION(S), IF ANY, FILED BEFORE THE PRIORITY APPLICATION(S)			
COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	DATE OF ISSUE (day, month, year)

I hereby claim the benefit under Title 35, United States Code, § 120/365 of any United States and PCT international application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. APPLICATION NUMBER	DATE OF FILING (day, month, year)	STATUS (patented, pending, abandoned)

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below:

U.S. PROVISIONAL APPLICATION NUMBER	DATE OF FILING (Day, Month, Year)

I hereby appoint the following attorney(s) and/or patent agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith:

Funk, Steven R.	Reg. No. 37,830	Crawford, Robert J.	Reg. No. 32,122	Maunu, LeRoy D.	Reg. No. 35,274
Hollingsworth, Mark A.	Reg. No. 38,491	Curtin, Eric J.	Reg. No. 47,511	Davis, Clara	Reg. No. 50,495

I hereby authorize them to act and rely on instructions from and communicate directly with the person/assignee/attorney/firm/organization who/which first sends/sent this case to them and by whom/which I hereby declare that I have consented after full disclosure to be represented unless/until I instruct Crawford PLLC.

Please direct all correspondence in this case to Crawford PLLC at the address indicated below:

Crawford PLLC
1270 Northland Drive, Suite 390
St. Paul, Minnesota 55120

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

2	Full Name Of Inventor	Family Name HANNUKSELA	First Given Name MISKA	Second Given Name
0	Residence & Citizenship	City TAMPERE	State or Foreign Country FINLAND	Country of Citizenship FINLAND
1	Post Office Address	Post Office Address KUKKANIITYNKATU 4 B	City TAMPERE	State & Zip Code/Country 33710/FINLAND
Signature of Inventor 201:			Date:	7 January 2003

§ 1.56 Duty to disclose information material to patentability.

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is canceled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is canceled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§ 1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) the closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made of record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

- A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application:

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

01/27/2006 HMARZI1 00000049 503581 11338934

01 FC:1011 300.00 DA
02 FC:1111 500.00 DA
03 FC:1311 200.00 DA
04 FC:1201 800.00 DA

PTO-1556
(5/87)

*U.S. Government Printing Office: 2002 — 489-267/69033

PTO/SB/06 (12-04)
Approved for use through 7/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875 Effective December 8, 2004

Application or Docket Number 11338931

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	150.00		N/A	300.00
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A	N/A	$250		N/A	$500
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	$100		N/A	$200
TOTAL CLAIMS (37 CFR 1.16(i))	16 minus 20 =	*	X$ 25 =		OR	X$50 =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	1 minus 3 =	* 4	X100 =			X200 =	
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))			+180=			+360=	
			TOTAL			TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X$ 25 =		OR	X$50 =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X100 =		OR	X200 =	
Application Size Fee (37 CFR 1.16(s))									
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					+180=		OR	+360=	
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)	OR	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X$ 25 =		OR	X$50 =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X100 =		OR	X200 =	
Application Size Fee (37 CFR 1.16(s))									
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					+180=		OR	+360=	
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Espacenet

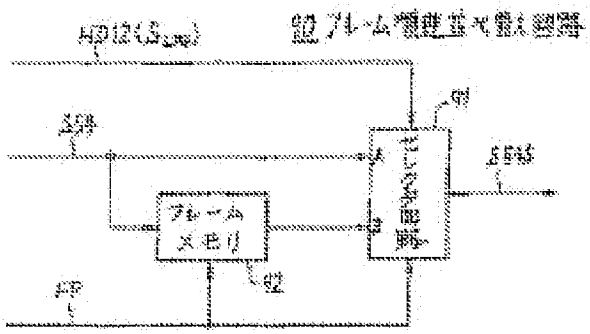
Bibliographic data: JP 8265694 (A)

ENCODED DATA DECODING METHOD AND DEVICE

Publication date:	1996-10-11
Inventor(s):	YONEMITSU JUN +
Applicant(s):	SONY CORP +
Classification:	**- international:** ***H04N5/91; H04N5/92; H04N7/24; H04N7/26;*** (IPC1-7): H04N5/91; H04N5/92; H04N7/24 **- European:**
Application number:	JP19950352516 19951230
Priority number (s):	JP19950352516 19951230
Also published as:	• JP 2874745 (B2)

Abstract of JP 8265694 (A)

PURPOSE: To rightly restore the encoded image data after an edition is performed by preventing the image for which a second inter-encoding is performed from being outputted by using the image belonging to an original preceding group as a predicted image when an edition flag is set. CONSTITUTION: By using the image belonging to an original preceding group as a predicted image, an edition flag SLPG is set to the group including the encoded data corresponding to the image for which a second inter-encoding which is possible to use the images existing before and after in the time in a display order as predicted images is performed.; A frame order reverse rearranging circuit 90 executes the processing rearranging from the frame order according to a decoding processing to the frame order based on an inut image signal and executes an edition reproduction processing by referring to the edition flag SLPG. Namely, when the edition flag SLPG is set, the image belonging to the original preceding frame group is used as the the predicted image, an interpolation frame is not outputted, the intraframe encoded just before is outputted and an encoding is performed.



Last updated: 26.04.2011 Worldwide Database 5.7.23; 93p

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平8－265694

(43)公開日　平成８年(1996)10月11日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ			技術表示箇所
Ｈ０４Ｎ	5/92			Ｈ０４Ｎ	5/92	Ｈ	
	5/91				5/91	Ｎ	
	7/24				7/13	Ｚ	

審査請求　有　　請求項の数４　ＦＤ　（全 21 頁）

(21)出願番号　特願平7－352516
(62)分割の表示　特願平2－119604の分割
(22)出願日　平成２年(1990)５月９日

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号
(72)発明者　米満　潤
東京都品川区北品川６丁目７番35号ソニー株式会社内
(74)代理人　弁理士　田辺　恵基

(54)【発明の名称】　符号化データ復号化方法及び符号化データ復号化装置

(57)【要約】
【課題】画像信号を複数のフレームを含む複数のグループに分割して符号化した場合、このグループを分断するような編集を行うと、編集後の画像を再生した際の画質が劣化する問題があつた。
【解決手段】編集フラグの有無を検出し、編集フラグが設定されていた場合には先行するグループに属する画像を予測画像に用いて、表示順上で時間的に前後にある画像を予測画像として用いることが可能なインター符号化によつて生成された画像を出力しないようにしたことにより、編集後の符号化画像データを正しく復元し得る符号化データ復号化方法及び装置を実現できる。

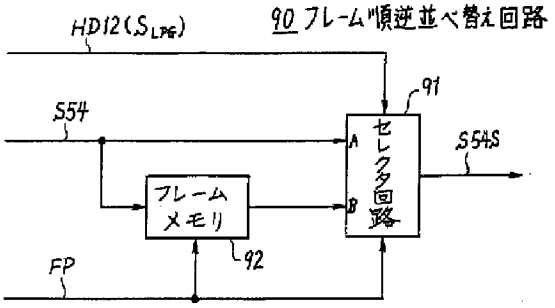


図１２　フレーム順逆並べ替え回路

【特許請求の範囲】
【請求項１】イントラ符号化によつて生成された第１の符号化データと、表示順上で時間的に前にある画像のみを予測画像として用いることが可能な第１のインター符号化によつて生成された第２の符号化データと、表示順上で時間的に前後にある画像を予測画像として用いることが可能な第２のインター符号化によつて生成された第３の符号化データとを含む符号化画像データが、それぞれいくつかの画像に対応する符号化データを含む複数のグループにグループ化され、上記符号化画像データは、元の先行するグループとは異なるグループに先行されるあるグループを少なくとも１つ含むように編集されてなり、上記あるグループは、上記元の先行するグループに属する画像を上記予測画像に用いて上記第２のインター符号化がなされた画像に対応する符号化データを含み、上記あるグループには編集フラグが設定されているような符号化画像データを復号化する符号化データ復号化方法において、
上記編集フラグの設定の有無を検出し、
上記編集フラグが設定されていた場合、上記元の先行するグループに属する画像を上記予測画像に用いて第２のインター符号化された画像を出力しないようにすることを特徴とする符号化データ復号化方法。
【請求項２】上記元の先行するグループに属する画像を上記予測画像に用いて第２のインター符号化された画像のかわりに、直前に復号化されたイントラ符号化された画像を出力することを特徴とする請求項１に記載の符号化データ復号化方法。
【請求項３】イントラ符号化によつて生成された第１の符号化データと、表示順上で時間的に前にある画像のみを予測画像として用いることが可能な第１のインター符号化によつて生成された第２の符号化データと、表示順上で時間的に前後にある画像を予測画像として用いることが可能な第２のインター符号化によつて生成された第３の符号化データとを含む符号化画像データが、それぞれいくつかの画像に対応する符号化データを含む複数のグループにグループ化され、上記符号化画像データは、元の先行するグループとは異なるグループに先行されるあるグループを少なくとも１つ含むように編集されてなり、上記あるグループは、上記元の先行するグループに属する画像を上記予測画像に用いて上記第２のインター符号化がなされた画像に対応する符号化データを含み、上記あるグループには編集フラグが設定されているような符号化画像データを復号化する符号化データ復号化装置において、
上記編集フラグの設定の有無を検出する手段と、
上記編集フラグが設定されていた場合、上記元の先行するグループに属する画像を上記予測画像に用いて第２のインター符号化された画像を出力しないようにする復号化手段とを具えることを特徴とする符号化データ復号化装置。
【請求項４】上記復号化手段は、上記元の先行するグループに属する画像を上記予測画像に用いて第２のインター符号化された画像のかわりに、直前に復号化されたイントラ符号化された画像を出力することを特徴とする請求項３に記載の符号化データ復号化装置。
【発明の詳細な説明】
【０００１】
【目次】以下の順序で本発明を説明する。
発明の属する技術分野
従来の技術（図１６～図１８）
発明が解決しようとする課題（図１９）
課題を解決するための手段
発明の実施の形態
（１）動画符号化／復号化装置の全体構成（図１～図５）
（２）実施例による動画符号化データの記録順序（図６～図９）
（３）実施例による動画符号化データのエデイツト処理（図６～図１１）
（４）実施例によるエデイツトされた符号化データの復号化方法（図８及び図１２）
（５）他の実施例（図１３～図１５）
発明の効果
【０００２】
【発明の属する技術分野】本発明は符号化データ復号化方法及び装置に関し、例えばエデイツト可能に符号化した符号化データを復号する際に適用して好適なものである。
【０００３】
【従来の技術】従来動画映像でなる映像信号をフレーム内符号化データ及びフレーム間符号化データに高能率符号化して、例えば光磁気デイスク構成のコンパクトデイスク（ＣＤ－ＭＯデイスク）に高密度記録し、当該記録された動画符号化データを必要に応じて検索して再生し得るようになされた記録再生装置が提案されている（特開昭63-1183号公報、特願平1-267049号）。
【０００４】すなわち、例えば図１６（Ａ）に示すように、時点ｔ＝t_1、t_2、t_3……において動画を構成する各画像ＰＣ１、ＰＣ２、ＰＣ３……をデイジタル符号化して、例えばＣＤ－ＭＯ記録再生装置でなる伝送系に伝送する場合、映像信号には時間の経過に従つて自己相関が大きい特徴がある点を利用して伝送すべき画像データを圧縮処理することにより伝送効率を高めるような処理をするもので、フレーム内符号化処理は画像ＰＣ１、ＰＣ２、ＰＣ３……を例えば画素データを所定の基準値と比較して差分を求めるような圧縮処理を実行し、かくして各画像ＰＣ１、ＰＣ２、ＰＣ３……について同一フレーム内における画素データ間の自己相関を利用して圧縮されたデータ量の画像データを伝送する。

【０００５】またフレーム間符号化処理は、図１６（Ｂ）に示すように、順次隣合う画像ＰＣ１及びＰＣ２、ＰＣ２及びＰＣ３……間の画素データの差分でなる画像データＰＣ１２、ＰＣ２３……を求め、これを時点ｔ＝t_1 における初期画像ＰＣ１についてフレーム内符号化処理された画像データと共に伝送する。かくして画像ＰＣ１、ＰＣ２、ＰＣ３……をそのすべての画像データを伝送する場合と比較して格段的にデータ量が少ないデイジタルデータに高能率符号化してＣＤ－ＭＯ記録再生装置に伝送することができる。

【０００６】かかる映像信号の符号化処理は、図１７に示す構成の動画符号化データ発生装置１において実行される。動画符号化データ発生装置１は入力映像信号ＶＤを前処理回路２において処理することにより片フイールド落し処理及び片フイールドライン間引き処理等の処理をした後、輝度信号及びクロマ信号を16画素（水平方向に）×16画素（垂直方向に）分のデータでなる伝送単位ブロツク（これをマクロブロツクと呼ぶ）データＳ１１に変換して画像データ符号化回路３に供給する。

【０００７】画像データ符号化回路３は予測符号化回路４において形成される予測現フレームデータＳ１２を受けてマクロブロツクデータＳ１１との差分を求めることによつてフレーム間符号化データを発生し（これをフレーム間符号化モードと呼ぶ）、又はマクロブロツクデータＳ１１と基準値データとの差分を求めることによりフレーム内符号化データを形成してこれを差分データＳ１３として変換符号化回路５に供給する。

【０００８】変換符号化回路５はデイスクリートコサイン変換回路で構成され、差分データＳ１３を直交変換することによつて高能率符号化してなる変換符号化データＳ１４を量子化回路６に与えることにより量子化画像データＳ１５を送出させる。かくして量子化回路６から得られる量子化画像データＳ１５は可変長符号化回路を含んでなる再変換符号化回路７において再度高能率符号化処理された後、伝送画像データＳ１６として伝送バツフアメモリ８に供給される。

【０００９】これに加えて量子化画像データＳ１５は予測符号化回路４において逆量子化、逆変換符号化処理されることより差分データに復号化された後予測前フレームデータを差分データによつて修正演算することにより新たな予測前フレームデータを保存すると共に、マクロブロツクデータＳ１１に基づいて形成される動き検出データによつて予測符号化回路４に保存されている予測前フレームデータを動き補償することにより予測現フレームデータを形成して画像データ符号化回路３に供給できるようになされ、これにより現在伝送しようとするフレーム（すなわち現フレーム）のマクロブロツクデータＳ１１と予測現フレームデータＳ１２との差分を差分データＳ１３として得るようになされている。

【００１０】図１７の構成において、図１６について上述した動画像を伝送する場合、先ず図１６（Ａ）の時点t_1 において画像ＰＣ１の画像データがマクロブロツクデータＳ１１として与えられたとき、画像データ符号化回路３はフレーム内符号化モードになつてこれをフレーム内符号化処理された差分データＳ１３として変換符号化回路５に供給し、これにより量子化回路６、再変換符号化回路７を介して伝送バツフアメモリ８に伝送画像データＳ１６を供給する。

【００１１】これと共に、量子化回路６の出力端に得られる量子化画像データＳ１５が予測符号化回路４において予測符号化処理されることにより、伝送バツフアメモリ８に送出された伝送画像データＳ１６を表す予測前フレームデータが前フレームメモリに保持され、続いて時点t_2 において画像ＰＣ２を表すマクロブロツクデータＳ１１が画像データ符号化回路３に供給されたとき、予測現フレームデータＳ１２に動き補償されて画像データ符号化回路３に供給される。

【００１２】かくして時点ｔ＝t_2 において画像データ符号化回路３はフレーム間符号化処理された差分データＳ１３を変換符号化回路５に供給し、これにより当該フレーム間の画像の変化を表す差分データが伝送画像データＳ１６として伝送バツフアメモリ８に供給されると共に、その量子化画像データＳ１５が予測符号化回路４に供給されることにより予測符号化回路４において予測前フレームデータが形成されると共に保存される。

【００１３】以下同様の動作が繰り返されることにより、画像データ符号化回路３がフレーム間符号化処理を実行している間、前フレームと現フレームとの間の画像の変化を表す差分データだけが伝送バツフアメモリ８に順次送出されることになる。伝送バツフアメモリ８はこのようにして送出されて来る伝送画像データＳ１６を一旦記憶し、伝送路９の伝送容量によつて決まる所定のデータ伝送速度で記憶された伝送画像データＳ１６を順次伝送データD_{TRANS} として引き出して伝送路９に伝送する。

【００１４】これと同時に伝送バツフアメモリ８は残留しているデータ量を検出して当該残留データ量に応じて変化する残量データＳ１７を量子化回路６にフイードバツクして残量データＳ１７に応じて量子化ステツプサイズを制御することにより、伝送画像データＳ１６として発生されるデータ量を調整することにより伝送バツフアメモリ８内に適正な残量（オーバーフロー又はアンダーフローを生じさせないようなデータ量）のデータを維持できるようになされている。

【００１５】因に伝送バツフアメモリ８のデータ残量が許容上限にまで増量して来たとき、残量データＳ１７によつて量子化回路６の量子化ステツプＳＴＰＳ（図１８）のステツプサイズを大きくすることにより、量子化回路６において粗い量子化を実行させることにより伝送画像データＳ１６のデータ量を低下させる。

【００１６】これとは逆に伝送バツフアメモリ８のデータ残量が許容下限値まで減量して来たとき、残量データＳ１７は量子化回路６の量子化ステツプＳＴＰＳのステツプサイズを小さい値になるように制御し、これにより量子化回路６において細かい量子化を実行させるようにすることにより伝送画像データＳ１６のデータ発生量を増大させる。

【００１７】

【発明が解決しようとする課題】ところでかかる構成の動画符号化データ発生装置１から送出される伝送データD_{TRANS}においては、図１９（Ａ）及び（Ｂ）に示すようにフレーム内符号化処理された完全フレーム内処理フレーム（以下これをイントラフレームと呼び、符号「Ａ」で表す）Ａ１、Ａ９、……と、フレーム内符号化処理された前フレーム予測処理フレーム（以下これを予測フレームと呼び、符号「Ｂ」で表す）Ｂ３、Ｂ５、Ｂ７、……及びそれらに応じた補間予測処理フレーム（以下これを補間フレームと呼び、符号「Ｃ」で表す）Ｃ２、Ｃ４、Ｃ６、……が、画像データＶＤの入力フレーム順に伝送されている。

【００１８】ところがこのような伝送データD_{TRANS}を受け、例えば補間フレームＣ２を復号化する際には、図１９（Ｃ）に示すようにイントラフレームＡ１及び予測フレームＢ３が必要になり、動画符号化データの復号化装置としてはイントラフレームＡ１及び予測フレームＢ３を受けるまでの間、補間フレームＣ２を遅延させるメモリやその制御回路が必要になり、その分回路構成が複雑になると共に遅延量が多大になることを避け得なかつた。

【００１９】このため伝送データD_{TRANS}を図１９（Ｃ）に示すように復号化処理に必要な順序で伝送することが考えられ、このような場合例えば伝送データD_{TRANS}はイントラフレームＡ１、Ａ９、……間の８フレーム分（Ａ１、Ｃ２、Ｂ３、Ｃ４、Ｂ５、Ｃ６、Ｂ７、Ｃ８）でなるフレーム群ＧＯＦ１、ＧＯＦ２がＣＤ－ＭＯデイスクの20セクタ分として記録するようになされている。

【００２０】ところがこのようなフレーム順でＣＤ－ＭＯデイスクに記録された動画符号化データについて、例えばフレーム群ＧＯＦ１を編集して書き換えるエデイツト処理を実行し、先頭から順次再生する際には、フレーム群ＧＯＦ１中の第８の補間フレームＣ８は、フレーム群ＧＯＦ１中の新たな第７の予測フレームＢ７とフレーム群ＧＯＦ２中の古い第１のイントラフレームＡ９に基づいて補間されることにより映像信号を得ることができず、結局エデイツト結果を正しく再生できないという問題があつた。

【００２１】本発明は以上の点を考慮してなされたもので、従来の問題を一挙に解決して、編集された動画像符号化データを、簡易な構成で正しく再生することができる符号化データ復号化方法及び装置を提案しようとするものである。

【００２２】

【課題を解決するための手段】かかる課題を解決するため本発明においては、イントラ符号化によつて生成された第１の符号化データと、表示順上で時間的に前にある画像のみを予測画像として用いることが可能な第１のインター符号化によつて生成された第２の符号化データと、表示順上で時間的に前後にある画像を予測画像として用いることが可能な第２のインター符号化によつて生成された第３の符号化データとを含む符号化画像データVD_{REC}が、それぞれいくつかの画像に対応する符号化データを含む複数グループＧＯＦにグループ化され、符号化画像データV_{REC}は、元の先行するグループＧＯＦとは異なるグループＧＯＦに先行されるあるグループＧＯＦを少なくとも１つ含むように編集されてなり、上記あるグループＧＯＦは、元の先行するグループＧＯＦに属する画像を予測画像に用いて第２のインター符号化がなされた画像に対応する符号化データを含み、上記あるグループＧＯＦには編集フラグS_{LPG}が設定されているような符号化画像データVD_{REC}（VD_{PB}）を復号化する符号化データ復号化装置２１において、編集フラグS_{LPG}の設定の有無を検出する手段９０と、編集フラグS_{LPG}が設定されていた場合、上記元の先行するグループＧＯＦに属する画像を上記予測画像に用いて第２のインター符号化された画像を出力しないようにする復号化手段とを設けるようにする。

【００２３】また本発明においては、復号化手段は、元の先行するグループＧＯＦに属する画像を予測画像に用いて第２のインター符号化された画像のかわりに、直前に復号化されたイントラ符号化された画像を出力するようにする。

【００２４】

【発明の実施の形態】以下図面について、本発明を動画符号化データ記録再生装置に適用した場合の実施例を詳述する。

【００２５】（１）動画符号化／復号化装置の全体構成図１及び図２において動画符号化／復号化装置２１は動画符号化装置２１Ａ及び動画復号化装置２１Ｂによつて構成され、動画符号化装置２１Ａは入力映像信号VD_{IN}を入力回路部２２において前処理した後、アナログ／デイジタル変換回路２３において16×16画素分の画素データでなる伝送単位ブロツクデータ、すなわちマクロブロツクＭＢの画素データでなる入力画像データＳ２１を画素データ処理系ＳＹＭ１に送り込むと共に、当該画素データ処理系ＳＹＭ１の各処理段においてマクロブロツクＭＢを単位として画素データが処理されるタイミングにおいて当該処理されるデータに対応する処理情報データがヘツダデータ処理系ＳＹＭ２を介して順次伝送されて行くようになされ、かくして画素データ及びヘツダデー

タがそれぞれ画素データ処理系ＳＹＭ１及びヘツダデータ処理系ＳＹＭ２において並列処理される。

【００２６】この実施例の場合、入力画像データＳ２１として順次送出されて来るマクロブロツクデータは、図３に示すような手法でフレーム画像データＦＲＭから抽出される。第１に入力映像信号VD_{IN}がＱＣＩＦの画サイズ（ 176× 144画素）でなる場合、先ず１枚のフレーム画像データＦＲＭは図３（Ａ１）に示すように１個（水平方向に）×３個（垂直方向に）のブロツクグループＧＯＢに分割され、各ブロツクグループＧＯＢが図３（Ｂ）に示すように11個（水平方向に）×３個（垂直方向に）のマクロブロツクＭＢを含むようになされ、各マクロブロツクＭＢは図３（Ｃ）に示すように16×16画素分の輝度信号データY_{00}～Y_{11}（それぞれ８×８画素分の輝度信号データでなる）及び輝度信号データY_{00}～Y_{11}の全画素データに対応する色信号データでなる色信号データC_b 及びC_r を含んでなる。

【００２７】これに対して第２に入力映像信号VD_{IN}がＣＩＦの画サイズ（252 ×288 画素）でなる場合、１枚のフレーム画像データＦＲＭは図３（Ａ２）に示すように２個（水平方向に）×６個（垂直方向に）のブロツクグループＧＯＢに分割され、各ブロツクグループＧＯＢが図３（Ｂ）に示すように11個（水平方向に）×３個（垂直方向に）のマクロブロツクＭＢを含むようになされ、各マクロブロツクＭＢは図３（Ｃ）に示すように16×16画素分の輝度信号データY_{00}～Y_{11}（それぞれ８×８画素分の輝度信号データでなる）及び輝度信号データY_{00}～Y_{11}の全画素データに対応する色信号データでなる色信号データC_b 及びC_r を含んでなる。

【００２８】かくしてマクロブロツクＭＢごとに送出される入力画像データＳ２１は動き補償回路２５に与えられ、動き補償回路２５はヘツダデータ処理系ＳＹＭ２に対して設けられている動き補償制御ユニツト２６から与えられる動き検出制御信号Ｓ２２に応動して予測前フレームメモリ２７の予測前フレームデータＳ２３と入力画像データＳ２１とを比較して動きベクトルデータMVD(x)及びMVD(y)を検出して動き補償制御ユニツト２６に第１のヘツダデータＨＤ１（図４）のデータとして供給すると共に、動き補償回路本体２５Ａにおいて予測前フレームデータＳ２３に対して動きベクトルデータMVD(x)及びMVD(y)分の動き補償をすることにより予測現フレームデータＳ２４を形成して現在処理しようとしている入力画像データＳ２１でなる現フレームデータＳ２５と共に画像データ符号化回路２８に供給する。

【００２９】ここで動き補償制御ユニツト２６は、図４に示すように、第１のヘツダデータＨＤ１として現在処理しているマクロブロツクごとに、フレーム画像データＦＲＭの伝送順序を表す伝送フレーム番号データTR Counterと、そのブロツクグループＧＯＢ（図３（Ａ１）、（Ａ２））を表すブロツクグループ番号データGOB address と、そのうちのマクロブロツクＭＢを表すマクロブロツク番号データMB addressとを付加することによつて順次画素データ処理系ＳＹＭ１の各処理段に伝送されて行くマクロブロツクＭＢを表示するようになされていると共に、当該処理対象マクロブロツクＭＢの処理ないし処理形式を表すフラグデータFLAGS と、当該マクロブロツクＭＢの動きベクトルデータMVD(x)及びMVD(y)と、その評価値を表す差分データΣ｜Ａ－Ｂ｜とを形成する。

【００３０】フラグデータFLAGS は図５に示すように、最大限１ワード（16ビツト）分のフラグをもち得るようになされ、第０ビツトには、当該処理対象マクロブロツクＭＢについて動き補償モードで処理すべきか否かを表す動き補償制御フラグMC on/off がセツトされる。またフラグデータFLAGS の第１ビツトには、当該処理対象マクロブロツクＭＢをフレーム間符号化モードで処理すべきであるか又はフレーム内符号化モードで処理すべきであるかを表すフレーム間／フレーム内フラグInter/Intra がセツトされる。

【００３１】またフラグデータFLAGS の第２ビツトには、動き補償回路２５のループフイルタ２５Ｂを使用するか否かを表すフイルタフラグFilter on/off が設定される。またフラグデータFLAGS の第３ビツトには、当該処理対象マクロブロツクに含まれるブロツクデータY_{00}～C_r （図３（Ｃ））を伝送すべきであるか否かを表す送信フラグCoded/Not-coded を設定できるようになされている。

【００３２】またフラグデータFLAGS の第４ビツトには、当該処理対象マクロブロツクＭＢを駒落しするか否かを表す駒落しフラグDrop frame flag を設定し得るようになされている。またフラグデータFLAGS の第５ビツトには、当該処理対象マクロブロツクＭＢを強制リフレツシユするか否かを表す強制リフレツシユフラグRefresh on/offを設定できるようになされている。

【００３３】またフラグデータFLAGS の第６ビツトには、マクロブロツクパワー評価フラグMBP appreciateを設定できるようになされている。また差分データΣ｜Ａ－Ｂ｜は、現フレームデータＳ２５の現在処理しようとするマクロブロツクデータＡと、予測前フレームデータＳ２３の検出用動きベクトルによつて補償されたマクロブロツクデータＢとの差分のうちの最小値を表し、これにより検出された動きベクトルの評価をなし得るようになされている。

【００３４】画像データ符号化回路２８はフレーム内符号化モードのとき動き補償回路２５から与えられる現フレームデータＳ２５をそのまま差分データＳ２６として変換符号化回路２９に供給し、これに対してフレーム間符号化モードのとき現フレームデータＳ２５の画素データと予測現フレームデータＳ２４の画素データとの差分でなる差分データＳ２６を変換符号化回路２９に供給する。

【００３５】ヘツダデータ処理系ＳＹＭ２には画像データ符号化回路２８に対応するようにフレーム間／フレーム内符号化制御ユニツト３０が設けられ、動き補償制御ユニツト２６から供給されるヘツダデータＨＤ１及び画像データ符号化回路２８から供給される演算データＳ３１に基づいて、画像データ符号化回路２８の符号化モードを指定するためのフレーム間／フレーム内フラグInter/Intra （図５）及び動き補償回路２５のループフイルタ２５Ｂの動作を制御するためのフイルタフラグFilter on/off （図５）とを得るのに必要なデータを演算して第２のヘツダデータＨＤ２としてフイルタ制御ユニツト３１に送出する。

【００３６】第２のヘツダデータＨＤ２は、図４に示すように、ヘツダデータＨＤ１を構成する伝送フレーム番号データTR Counter～差分データΣ｜Ａ－Ｂ｜をそのまま引き継ぐと共に、フイルタ制御ユニツト３１においてフレーム間／フレーム内符号化モード切換信号Ｓ３３及びフイルタオン／オフ信号Ｓ３４を形成するために必要なパワーデータΣ（A)^2（Ｌ）及びΣ（A)^2（Ｈ）、Σ（Ａ－Ｂ)^2（Ｌ）及びΣ（Ａ－Ｂ)^2（Ｈ）、Σ（Ａ－ＦＢ)^2（Ｌ）及びΣ（Ａ－ＦＢ)^2（Ｈ）、Σ（Ａ）をフレーム間／フレーム内符号化制御ユニツト３０において付加されるようになされている。

【００３７】ここで、パワーデータΣ（A)^2（Ｌ）及びΣ（A)^2（Ｈ）は現フレームデータＳ２５のマクロブロツク画素データＡの２乗和の下位ビツト及び上位ビツトを表し、パワーデータΣ（Ａ－Ｂ)^2（Ｌ）及びΣ（Ａ－Ｂ)^2（Ｈ）は現フレームデータＳ２５のマクロブロツク画素データＡとループフイルタ２５Ｂを介さずに形成された予測現フレームデータＳ２４のマクロブロツク画素データＢとの差分Ａ－Ｂの２乗和の下位ビツト及び上位ビツトを表し、パワーデータΣ（Ａ－ＦＢ)^2（Ｌ）及びΣ（Ａ－ＦＢ)^2（Ｈ）は現フレームデータＳ２５のマクロブロツク画素データＡとループフイルタ２５Ｂを介して形成された予測現フレームデータＳ２４のマクロブロツク画素データＦＢとの差分Ａ－ＦＢの２乗和の下位ビツト及び上位ビツトを表し、パワーデータΣ（Ａ）は現フレームデータＳ２５のマクロブロツク画素データＡの和を表し、それぞれ処理するデータの大きさを評価するためにデータ量をパワー値として表現したもの（２乗和は符号に無関係な値として求めた）である。

【００３８】フイルタ制御ユニツト３１は、フレーム間／フレーム内符号化制御ユニツト３０から渡された第２のヘツダデータＨＤ２と、伝送バツフアメモリ３２から供給される残量データＳ３２とに基づいて、画像データ符号化回路２８に対してフレーム間／フレーム内符号化モード切換信号Ｓ３３を送出すると共に、ループフイルタ２５Ｂに対してフイルタオン／オフ信号Ｓ３４を送出すると共に、当該フイルタオン／オフ信号Ｓ３４の内容を表すフイルタフラグFilter on/off を第２のヘツダデータＨＤ２に付加して第３のヘツダデータＨＤ３としてスレシヨルド制御ユニツト３５に渡す。ここでフイルタ制御ユニツト３１は第１に、フレーム間符号化処理をした場合の伝送データ量の方がフレーム内符号化処理をした場合の伝送データ量より大きくなつたとき画像データ符号化回路２８をフレーム内符号化モードに制御する。

【００３９】またフイルタ制御ユニツト３１は第２に、フレーム間符号化モードで処理をしている状態においてループフイルタ２５Ｂにおける処理を受けた予測現フレームデータＳ２４より当該処理を受けない予測現フレームデータＳ２４の方が差分値が小さい場合には、フイルタオン／オフ信号Ｓ３４によつてフイルタリング動作をさせないようにループフイルタ２５Ｂを制御する。

【００４０】またフイルタ制御ユニツト３１は第３に、強制リフレツシユモードになつたとき、フレーム間／フレーム内符号化モード切換信号Ｓ３３によつて画像データ符号化回路２８をフレーム内符号化モードに切り換える。さらにフイルタ制御ユニツト３１は第４に、伝送バツフアメモリ３２から供給される残量データＳ３２に基づいて伝送バツフアメモリ３２がオーバーフローするおそれがある状態になつたとき、これを検出して駒落し処理をすべきことを命令するフラグを含んでなる第３のヘツダデータＨＤ３をスレシヨルド制御ユニツト３５に送出する。

【００４１】かくして画像データ符号化回路２８は現フレームデータＳ２５と予測現フレームデータＳ２４との差分が最も小さくなるようなモードで符号化してなる差分データＳ２６を変換符号化回路２９に供給する。

【００４２】第３のヘツダデータＨＤ３は、図４に示すように、ヘツダデータＨＤ２から伝送フレーム番号データTR Counter～動きベクトルデータMVD(x)及びMVD(y)を引き継ぐと共に、フイルタ制御ユニツト３１においてブロツクデータY_{00}～C_r に対応する６ビツト分のフイルタフラグFilter on/off を付加される。

【００４３】変換符号化回路２９はデイスクリートコサイン変換回路でなりデイスクリートコサイン変換後の係数値を６個のブロツクY_{00}、Y_{01}、Y_{10}、Y_{11}、C_b、C_rごとにジグザグスキヤンしてなる変換符号化データＳ３５として伝送ブロツク設定回路３４に送出する。

【００４４】伝送ブロツク設定回路３４は変換符号化データＳ３５として送出されて来る６個のブロツクデータY_{00}～C_r （図３（Ｃ））について、それぞれ先頭の係数データからｎ個までの２乗和を演算して当該演算結果をパワー検出データＳ３６としてスレシヨルド制御ユニツト３５に渡す。

【００４５】このときスレシヨルド制御ユニツト３５は各ブロツクデータY_{00}～C_r ごとにパワー検出データＳ３６を所定のスレシヨルドと比較し、パワー検出データＳ３６が当該スレシヨルドより小さいとき当該ブロツクデータの伝送を許容せず、これに対して大きいとき許容

することを表す６ビット分の伝送可否データＣＢＰＮを形成してこれをフイルタ制御ユニット３１から渡された第３のヘッダデータＨＤ３に付加して第４のヘッダデータＨＤ４として量子化制御ユニット３６に渡すと共に、伝送ブロック設定回路３４から対応するブロックデータY_{00}～C_rを量子化回路３７に送信ブロックパターン化データＳ３７として送出させる。

【００４６】ここで第４のヘッダデータＨＤ４は図４に示すように、ヘッダデータＨＤ３の伝送フレーム番号データTR Counter～フイルタフラグFilter on/off をそのまま引き継ぐと共に、スレショルド制御ユニット３５においてブロックY_{00}～C_rに対応して発生する６ビット分の送信可否フラグＣＢＰＮが付加される。

【００４７】量子化制御ユニット３６はスレショルド制御ユニット３５から渡された第４のヘッダデータＨＤ４と、伝送バッファメモリ３２から送出される残量データＳ３２とに基づいて、量子化ステップサイズ決定処理を実行して得られる量子化ステップサイズ制御信号Ｓ３８を量子化回路３７に与え、これにより量子化回路３７をマクロブロックＭＢに含まれるデータに適応した量子化ステップサイズで量子化処理させ、その結果量子化回路３７の出力端に得られる量子化画像データＳ３９を可変長符号化回路３８に供給させる。

【００４８】これと共に量子化制御ユニット３６は、図４に示すように、第５のヘッダデータＨＤ５として、ヘッダデータＨＤ４に基づいてブロックデータ　Y_{00}～C_r　（図３（Ｃ））にそれぞれ対応するフラグデータFLAGS 及び動きベクトルデータMVD(x)及びMVD(y)に分離してこれを直列に配列させたデータを形成して可変長符号化回路３８及び逆量子化回路４０に渡す。

【００４９】ここで、ヘッダデータＨＤ５は、図４に示すように、ヘッダデータＨＤ４のうち伝送フレーム番号データTR Counter～マクロブロック番号データMB addressをそのまま引き継ぐと共に、量子化制御ユニット３６において量子化サイズデータQNT と、ブロックデータY_{00}～C_rに対するフラグデータFLAGS 、動きベクトルデータMVD(x)及びMVD(y)を付加する。

【００５０】可変長符号化回路３８はヘッダデータＨＤ５及び量子化画像データＳ３９を可変長符号化処理して伝送画像データＳ４０を形成し、これを伝送バッファメモリ３２に供給する。可変長符号化回路３８はブロックデータY_{00}～C_rを可変長符号化する際に、対応するフラグデータFLAGS に基づいて「駒落し」、又は「送信不可」が指定されているとき、当該ブロックデータを伝送画像データＳ４０として送出させずに捨てるような処理をする。

【００５１】伝送バッファメモリ３２は伝送画像データＳ４０を溜め込んで行くと共に、これを所定の伝送速度で読み出してマルチプレクサ４１において音声データ発生装置４２から送出される伝送音声データＳ４１と合成して動画符号化データVD_{REC}としてＣＤ－ＭＯ装置に送出する。

【００５２】逆量子化回路４０は量子化回路３７から送出される量子化画像データＳ３９をヘッダデータＨＤ５に基づいて逆量子化した後、当該逆量子化データＳ４２を逆変換符号化回路４３に供給することにより逆変換符号化データＳ４３に変換させた後デコーダ回路４４に供給させ、かくして伝送画像データＳ４０として送出された画像情報を表す符号化差分データＳ４４を予測前フレームメモリ２７に供給させる。このとき予測前フレームメモリ２７は、符号化差分データＳ４４を用いてそれまで保存していた予測前フレームデータを修正演算して新たな予測前フレームデータとして保存する。

【００５３】かくして図１の構成の動画符号化装置２１Ａによれば、ヘッダデータ処理系ＳＹＭ２から供給されるヘッダ情報に基づいて画素データ処理系ＳＹＭ１において画素データがマクロブロック単位でパイプライン処理されて行くのに対して、これと同期するようにヘッダデータ処理系ＳＹＭ２においてヘッダデータを受け渡して行くようにすることにより、ヘッダデータ処理系ＳＹＭ２の各処理段において必要に応じてヘッダデータを付加又は削除することにより画素データを必要に応じて適応処理できる。

【００５４】動画復号化装置２１Ｂは図２に示すように、ＣＤ－ＭＯ装置から再生される動画符号化データVD_{PB}をデマルチプレクサ５１を介して伝送バッファメモリ５２に受けると共に、伝送音声データＳ５１を音声データ受信装置５３に受ける。

【００５５】伝送バッファメモリ５２に受けた画像データは可変長逆変換回路５４において受信画像データＳ５２及びヘッダデータＨＤ１１に分離され、逆量子化回路５５において逆量子化データＳ５３に逆量子化された後、逆変換符号化回路５６においてデイスクリート逆変換処理されて逆変換符号化データＳ５４に逆変換される。

【００５６】この逆変換符号化データＳ５４は逆量子化回路５５において形成されたヘッダデータＨＤ１２と共にデコーダ回路５７に与えられ、符号化差分データＳ５５としてフレームメモリ５８に蓄積される。

【００５７】かくしてフレームメモリ５８には符号化差分データＳ５５に基づいて伝送されてきた画像データが復号化され、当該復号化画像データＳ５６がデイジタル／アナログ変換回路５９においてアナログ信号に変換された後、出力回路部６０を介して出力映像信号VD_{OUT}として送出される。

【００５８】（２）実施例による動画符号化データの記録順序

図１及び図２との対応部分に同一符号を付して示す図６において、７０は全体として本発明による符号化データ復号化方法を適用した動画符号化データ記録再生装置を

示す。この動画符号化データ記録再生装置７０の場合、入力映像信号VD_{IN}が上述した動画符号化／復号化装置２１を通じて高能率符号化され、この結果得られる動画符号化データVD_{REC}がＣＤ－ＭＯ装置７１に入力されてＣＤ－ＭＯデイスク（図示せず）に記録される。

【００５９】逆にＣＤ－ＭＯ装置７１から得られる再生信号が動画符号化データVD_{PB}として、動画符号化／復号化装置２１に入力され、この結果動画符号化データVD_{PB}を復号化して得られる出力映像信号VD_{OUT}が送出される。この動画符号化データ記録再生装置７０の場合、動画符号化／復号化装置２１はバス７２を通じてＣＰＵ（中央処理ユニツト）を含む記録再生制御回路７３に接続され、この記録再生制御回路７３によつて入力映像信号VD_{IN}の符号化及び動画符号化データVD_{PB}の復号化が制御される。

【００６０】これに加えてＣＤ－ＭＯ装置７１はＳＣＳＩ(small computer system interface)を内蔵し、ＳＣＳＩバス７４、ＳＣＳＩ制御回路７５及びバス７２を通じて、記録再生制御回路７３に接続され、これにより記録再生制御回路７３によつて記録再生動作が制御される。

【００６１】ここで動画符号化／復号化装置２１から送出される動画符号化データVD_{REC}は、図７に示すような階層（レイヤ）構造を有するフオーマツトでＣＤ－ＭＯ装置７１に入力され、またＣＤ－ＭＯ装置７１から同様のフオーマツトで動画符号化データVD_{PB}として動画符号化／復号化装置２１に入力される。

【００６２】すなわち動画符号化データVD_{REC}及びVD_{PB}においては、フレーム群レイヤとして入力画像信号VD_{IN}の８フレーム分に対応したデータを１フレーム群ＧＯＦとして、当該１フレーム群の開始を表すフレーム群スタートコード(GOFSC)、直前のＧＯＦとの連続関係を表すリンクフラグ(LPG)、伝送するフレームの水平、垂直サイズや水平及び垂直方向の画素数比等を表すデータ(HORSIZE、VERSIZE、HVPRAT)、伝送フレームのレートを表すデータ(RATE)、１フレーム分の画像データでなるピクチヤレイヤのデータ（P.data）の８フレーム分及びスタツフイングビツト(TSB)から構成されている（図７（Ａ））。

【００６３】ピクチヤレイヤのデータ（P.data）の１フレーム分は、１フレームの開始を表すフレームスタートコード(PSC)、フレーム番号(TR)、拡張情報を表すデータ(PEI、PSPARE)及びブロツク単位の画像データでなるブロツクグループレイヤのデータ(GOB data)の１フレーム分から構成されている（図７（Ｂ））。

【００６４】ブロツクグループレイヤのデータ(GOB data)の１ブロツクグループ分は、１ブロツクグループの開始を表すブロツクグループスタートコード(GBSC)、ブロツクグループのアドレスデータ（GN）、ブロツクグループ単位の再量子化ステツプサイズに関するデータ(GQUANT)、拡張情報を表すデータ(GEI、GSPARE)及びマクロブロツクレイヤのデータ(MB data)の１ブロツクグループ分から構成されている（図７（Ｃ））。

【００６５】マクロブロツクレイヤのデータ(MB data)の１マクロブロツク分は、マクロブロツクのアドレスを表すデータ(MBA)、マクロブロツクのタイプを表すデータ(MTYPE)、マクロブロツクにおける再量子化ステツプサイズのデータ(MQUANT)、マクロブロツク毎の動きベクトルのデータ(MVD1、MVD2)、マクロブロツク内のブロツクパターンのデータ(CBP)及びブロツクレイヤのデータ(Block data)の１マクロブロツク分から構成されている（図７（Ｄ））。

【００６６】ブロツクレイヤのデータ(Block data)の１ブロツク分は、所定数の係数データ(TCOEF)とブロツクレイヤの終わりを表すデータ(EOB)から構成されている（図７（Ｅ））。

【００６７】ここでこの実施例による動画符号化装置２１Ａにおいては、図８（Ａ）に示す従来同様の入力画像信号VD_{IN}に基づくフレーム記録順序に代え、図８（Ｂ）に示すように動画復号化装置２１Ｂ側の復号化処理に応じたフレーム記録順序でなる動画符号化データVD_{REC}をＣＤ－ＭＯ装置７１に送出し記録するようになされ、ＣＤ－ＭＯ装置７１から再生された動画符号化データVD_{PB}がこのフレーム記録順序で動画復号化装置２１Ｂに入力される。

【００６８】このようにすれば、例えば補間フレームＣ２を復号化する際、復号化に必要なイントラフレームＡ１及び予測フレームＢ３がすでに入力され、また例えば補間フレームＣ４を復号化する際、復号化に必要な予測フレームＢ３及びＢ５がすでに入力され、これにより動画復号化装置２１Ｂは直ちに補間フレームＣ２又はＣ４の復号化処理を実行することができる。

【００６９】ここでこの実施例の動画符号化装置２１Ａの場合、図９に示すように、動き補償回路２５に内蔵されたフレーム順並べ替え回路８０を用いて、入力画像信号VD_{IN}に基づくフレーム記録順序を上述のように復号化処理に応じたフレーム記録順序に並べ替えるようになされている。

【００７０】このフレーム順並べ替え回路８０においては、第１～第３の１フレーム遅延回路８１、８２及び８３を有して構成されており、フレーム順の並べ替え処理に加えて、動きベクトルの検出処理を実行し得るようになされている。すなわち入力画像信号VD_{IN}は入力回路部２２及びアナログデイジタル変換回路２３において所定の処理が施され、この結果得られる入力画像データＳ２１が第１の１フレーム遅延回路８１に入力される。

【００７１】この第１の１フレーム遅延回路８１から送出された第１の遅延データS_{D1}は、第２の１フレーム遅延回路８２に入力されると共に、第１のフレーム選択回路８４の第１の入力端ａに入力される。また第２の１フ

レーム遅延回路８２から送出される第２の遅延データS_{D2}は第３の１フレーム遅延回路８３を通じ、さらに１フレーム分遅延され第３の遅延データS_{D3}として第１のフレーム選択回路８４の第２の入力端ｂに入力される。

【００７２】これによりこのフレーム順並べ替え回路８０においては、１フレーム毎のタイミングで順次入力画像データＳ２１を入力すると共に、これに同期したフレームパルスＦＰのタイミングで順次第１のフレーム選択回路８４の第１又は第２の入力端ａ又はｂを選択制御することにより、フレーム順の並べ替え処理を実行し得るようになれている。

【００７３】なお入力画像データＳ２１及び第１の遅延データS_{D1}は第１の動きベクトル検出回路８５に入力され、この結果得られる入力画像データＳ２１及び第１の遅延データS_{D1}間の動きベクトルデータD_{MV1}及び差分データD_{DR1}が動き補償ユニツト２６に送出される。

【００７４】また第２及び第３の遅延データS_{D2}及びS_{D3}が第２のフレーム選択回路８６の第１及び第２の入力端ａ及びｂに入力され、フレームパルスＦＰのタイミングで何れか一方が選択されて第２の動きベクトル検出回路８７に入力される。この第２の動きベクトル検出回路８７には、これに加えて第１の遅延データS_{D1}が入力されており、この結果得られる第１の遅延データS_{D1}及び第２又は第３の遅延データS_{D2}又はS_{D3}間の動きベクトルデータD_{MV2}及び差分データD_{DR2}が動き補償ユニツト２６に送出される。かくして入力画像信号VD_{IN}について高能率符号処理して動画符号化データVD_{REC}を得る際に、入力画像信号VD_{IN}に基づくフレーム順序に代えて、復号化処理に応じたフレーム順序に並べ替えるようにしたことにより、復号化処理側の回路構成及び制御を簡略かつ効率化し得る動画符号化データ伝送方法を実現できる。

【００７５】（３）実施例による動画符号化データのエデイツト処理

ここでこの動画符号化データ記録再生装置７０の記録再生制御回路７３は、外部から入力されたエデイツト命令に応動して、図１０に示すエデイツト処理手順ＲＴ０を実行し、これにより図１１に示すように、ＣＤ－ＭＯデイスク上の20セクタ毎に１フレーム群分記録された動画符号化データVD_{REC}を１フレーム群単位で書換え、かくしてエデイツト処理を実行するようになされている。

【００７６】なおこの実施例の場合動画符号化データVD_{IN}においては、図８（Ｃ）に示すようにイントラフレームから続く補間フレーム及び予測フレームＡ１、Ｃ２、Ｂ３、Ｃ４、Ｂ５、Ｃ６、Ｂ７、Ｃ８の８フレーム分を１フレーム群ＧＯＦ１、ＧＯＦ２、……とした従来の方法に代え、イントラフレームＡ１の直前の補間フレームＣ０に続くイントラフレーム、補間フレーム及び予測フレームＣ０、Ａ１、Ｃ２、Ｂ３、Ｃ４、Ｂ５、Ｃ６、Ｂ７を１フレーム群ＧＯＦ１１、ＧＯＦ１２、……として伝送するようになされている。

【００７７】このようにして１フレーム群ＧＯＦ１１、ＧＯＦ１２、……内のフレームが、他のフレーム群ＧＯＦ１１、ＧＯＦ１２、……に含まれてしまう不都合を未然に防止し得るようになされている。実際上記録再生制御回路７３は、図１０に示すエデイツト処理手順ＲＴ０から入つて次のステツプＳＰ１においてエデイツト命令の解析処理を実行する。

【００７８】ここで例えば図１１に示す第３及び第４のフレーム群ＧＯＦ３及びＧＯＦ４の40セクタ分について、新たな動画符号化データVD_{REC}と書き換えることが指示されると、記録再生制御回路７３は次のステツプＳＰ２に移る。

【００７９】このステツプＳＰ２において、記録再生制御回路７３はＳＣＳＩ制御回路７５にエデイツト命令に応じた制御命令を送出し、これによりＳＣＳＩバス７４を通じてＣＤ－ＭＯデイスクの書き換え制御を実行する。なおこのときバス７２を通じた記録再生制御回路７３の制御によつて、新たな２フレーム群ＧＯＦ３Ｎ、ＧＯＦ４Ｎ分の入力映像信号VD_{IN}が動画符号化／復号化装置２１を通じて動画符号化データVD_{REC}としてＣＤ－ＭＯ装置７１に入力される。

【００８０】続いて記録再生制御回路７３は次のステツプＳＰ３を実行し、書換え処理が終了したか否かを判断し、ここで否定結果を得るとステツプＳＰ２に戻りＣＤ－ＭＯデイスクの書き換え制御を継続し、やがて肯定結果を得るとステツプＳＰ４に移る。このステツプＳＰ４において記録再生制御回路７３は、エデイツトしたＣＤ－ＭＯデイスクの先頭セクタ（この実施例の場合、第40セクタでなる）の内容を読む。

【００８１】これにより記録再生制御回路７３は次のステツプＳＰ５において、先頭セクタの先頭から25ビツト目に存在するフレーム群レイヤのリンクフラグ(LPG) をエデイツトフラグとして設定し、これをＣＤ－ＭＯデイスクの読み出したセクタ位置に書き込む。

【００８２】続いて記録再生制御回路７３はステツプＳＰ６において、エデイツトしたＣＤ－ＭＯデイスクの最終セクタに続くセクタ（この実施例の場合、第80セクタでなる）の内容を読む。これにより記録再生制御回路７３は次のステツプＳＰ７において、当該セクタの先頭から25ビツト目に存在するフレーム群レイヤのリンクフラグ(LPG) を上述と同様にエデイツトフラグとして設定し、これをＣＤ－ＭＯデイスクの読み出したセクタ位置に書き込み、次のステツプＳＰ８において当該エデイツト処理手順ＲＴ０を終了する。

【００８３】実際上このようにしてエデイツトされてＣＤ－ＭＯデイスク上に記録された動画符号化データVD_{REC}は、記録再生制御回路７３の制御によつて読み出され、この結果再生信号として得られる動画符号化データVD_{PB}が動画符号化／復号化装置２１に入力される。

【００８４】（４）実施例によるエデイツトされた符号化データの復号化方法
この実施例の場合動画復号化装置２１Ｂにおいては、図１２に示すようにデコーダ回路５７に含まれるフレーム順逆並べ替え回路９０を用いて、復号化処理に応じたフレーム順序から入力画像信号VD_{IN}に基づくフレーム順序に並べ替える逆並べ替え処理を実行すると共に、フレーム群レイヤのリンクフラグ(LPG) に設定されたエデイツトフラグを参照してエデイツト再生処理を実行するようになされている。
【００８５】すなわち逆変換符号化回路５６から送出される逆変換符号化データＳ５４は、デコーダ回路５７のセレクタ回路９１の第１の入力端Ａに直接入力されると共に、フレームメモリ９２を通じて例えば２フレーム分遅延されて第２に入力端Ｂに入力される。
【００８６】このセレクタ回路９１及びフレームメモリ９２はそれぞれフレームパルスＦＰのタイミングで動作し、これにより復号化処理に応じたフレーム順序（図８（Ｂ））から入力画像信号VD_{IN}に基づくフレーム順序（図８（Ａ））への逆並べ替え処理を実行するようになされている。
【００８７】なおこのセレクタ回路９１には、逆量子化回路５５において形成されたヘツダデータＨＤ１２のうちリンクフラグ(LPG) に設定されたエデイツトフラグに応じたエデイツトフラグ信号S_{LPG}が入力され、このエデイツトフラグ信号S_{LPG}が設定されていることを表すときのみセレクト動作を中断し、次に到来するフレームをそのまま出力する。
【００８８】このようにして例えばエデイツトされた新たなフレーム群ＧＯＦＮ３、ＧＯＦＮ４中のフレームデータに対して、古いフレーム群ＧＯＦ１、ＧＯＦ２、ＧＯＦ５、……中のフレームデータが混入して再生画像に乱れが生じるおそれを未然に防止し得るようになれている。因に図８（Ｄ）に示すフレーム群ＧＯＦ１２にエデイツトフラグが設定されている場合には、逆並べ替え後の第８の補間フレームＣ８に代えて、イントラフレームＡ９が２フレーム分出力される。
【００８９】以上の構成によれば、エデイツトされたフレーム群及びエデイツト直後のフレーム群のリンクフラグ(LPG) にエデイツトフラグS_{LPG}が設定されてなる編集後の動画符号化データを復号化する際に、エデイツトフラグS_{LPG}の有無を検出し、エデイツトフラグS_{LPG}が設定されている場合には元の先行するフレーム群に属する画像を予測画像に用いて補間フレームを出力せずに、直前に復号化したイントラフレームを出力するようにして復号化するようにしたことにより、編集後の動画符号化データを正しく再生することができる。
【００９０】（５）他の実施例
（５－１）上述の実施例においては、記録された動画符号化データについてフレーム群単位でエデイツト処理を実行する際、エデイツトされたフレーム群及びエデイツト直後のフレーム群のリンクフラグ(LPG) にエデイツトフラグを設定した場合について述べたが、これに代え、例えばエデイツト処理したフレーム群を記録再生制御回路７３が記憶している場合には、エデイツト直後のフレーム群のリンクフラグ(LPG) にエデイツトフラグを設定するのみで良好にエデイツト処理を実行し得る。
【００９１】（５－２）また上述の実施例においては、記録された動画符号化データについてフレーム群単位でエデイツト処理を実行する際、エデイツトされたフレーム群及びエデイツト直後のフレーム群のリンクフラグ(LPG) にエデイツトフラグを設定した場合について述べたが、これに代え、動画符号化装置２１Ａ側でピクチヤレイヤのフレーム番号(TR)に、記録再生制御回路７３で発生した所定の乱数から始まる連番を順次付加し、エデイツト再生処理時にこのフレーム番号(TR)の不連続を検出したタイミングで上述したエデイツトフラグ信号S_{LPG}と同様の信号を発生するようにすれば、上述の実施例と同様の効果を実現できる。
【００９２】因に、この場合フレーム番号(TR)の不連続は、図１３に示すような不連続検出回路９５で検出される。すなわちこの不連続検出回路９５においては、上述のフレーム順逆並べ替え回路９０に併設され、逆変換符号化データＳ５４に含まれるピクチヤレイヤのフレーム番号(TR)が比較回路９６及びラツチ回路９７に入力される。
【００９３】このラツチ回路９７はフレームパルスＦＰのタイミングでラツチ動作を実行し、この結果１フレーム分遅延したフレーム番号(TR)が加算回路９８に入力されて値「１」が加算され、比較フレーム番号C_{TR}として比較回路９６に入力される。これにより比較回路９６は、フレーム番号(TR)及び比較フレーム番号C_{TR}の値を比較し、両者が不一致のとき論理「Ｈ」レベルでなる不一致検出信号を発生し、これをアンド回路９９に送出する。
【００９４】アンド回路９９にはこれに加えて、先頭フレーム群の再生処理のとき論理「Ｈ」レベルを有する先頭フレーム群信号ＧＯＦＦが反転回路１００を通じて反転して入力されており、これにより先頭フレーム群の再生処理のとき、不一致検出信号を論理「Ｌ」レベルに制御し、それ以外のとき不一致検出信号に応じた論理レベルを有するエデイツトフラグ信号S_{LPG}をフレーム順逆並べ替え回路９０のセレクタ回路９１に送出する。
【００９５】（５－３）さらに上述の実施例においては、動画符号化データについてフレーム群単位でエデイツト処理を実行する際、エデイツトフラグやフレーム番号(TR)を用いてエデイツトされたフレーム群を識別するようにしたが、フレーム群の各補間フレームＣについてフレーム群内のイントラフレームＡ及び予測フレームＢのみを用いて補間し、フレーム群内で各フレームの関係

が完結するようにすれば、画質について多少の劣化が生じても容易かつ自在にエデイツト処理を実行し得る動画符号化データ記録方法を実現できる。

【００９６】（５－４）さらに上述の実施例においては、イントラフレームＡ、予測フレームＢ及び補間フレームＣが、図８（Ａ）に示すように配置された動画符号化データのフレーム順序を並べ替えるようにした場合について述べたが、動画符号化データのフレーム配置はこれに限らず、図１４（Ａ）や図１５（Ａ）に示すような場合でも、要は図１４（Ｂ）や図１５（Ｂ）に示すように復号化側の処理順に応じたフレーム順に並べ替えて伝送するようにすれば、上述の実施例と同様の効果を実現できる。

【００９７】またこの場合図１４（Ｃ）や図１５（Ｃ）に示すフレーム群ＧＯＦ１、ＧＯＦ２、……の配置に代え、図１４（Ｄ）や図１５（Ｄ）に示すようなフレーム群ＧＯＦ２１、ＧＯＦ２２、……やＧＯＦ３１、ＧＯＦ３２、……の配置にすれば、エデイツト処理についても上述の実施例と同様の効果を実現できる。

【００９８】（５－５）さらに上述の実施例においては、映像信号を高能率符号化してＣＤ－ＭＯデイスクに記録し再生する場合について述べたが、記録媒体はこれに限らず、他の光デイスクや磁気デイスク、磁気テープ等に広く適用して好適なものである。

【００９９】（５－６）さらに上述の実施例においては、映像信号を高能率符号化してＣＤ－ＭＯデイスクに記録し再生する場合について述べたが、本発明はこれに限らず、映像信号を高能率符号化して伝送する動画符号化データ伝送方法に広く適用して好適なものである。

【０１００】

【発明の効果】上述のように本発明によれば、編集フラグの有無を検出し、編集フラグが設定されていた場合には先行するグループに属する画像を予測画像に用いて、表示順上で時間的に前後にある画像を予測画像として用いることが可能なインター符号化によつて生成された画像を出力しないようにしたことにより、編集後の符号化画像データを正しく復元し得る符号化データ復号化方法及び装置を実現できる。

【図面の簡単な説明】

【図１】実施例による動画符号化装置の構成を示すブロツク図である。

【図２】実施例による動画復号化装置の構成を示すブロツク図である。

【図３】フレーム画像データの構成を示す略線図である。

【図４】ヘツダデータ処理系の詳細構成を示すブロツク図である。

【図５】フラグデータの構成を示す略線図である。

【図６】実施例による動画符号化データ記録再生装置の構成を示すブロツク図である。

【図７】記録再生データのフオーマツトの説明に供する略線図である。

【図８】実施例による動画符号化データの記録順序の説明に供する略線図である。

【図９】フレーム順並べ替え回路の構成を示すブロツク図である。

【図１０】エデイツト処理の説明に供するフローチヤートである。

【図１１】ＣＤ－ＭＯデイスクの記録領域の説明に供する略線図である。

【図１２】フレーム順逆並べ替え回路の構成を示すブロツク図である。

【図１３】他の実施例による不連続検出回路を示すブロツク図である。

【図１４】他の実施例による動画符号化データの記録順序の説明に供する略線図である。

【図１５】他の実施例による動画符号化データの記録順序の説明に供する略線図である。

【図１６】フレーム内／フレーム間符号化処理の説明に供する略線図である。

【図１７】従来の動画符号化データ発生装置を示すブロツク図である。

【図１８】その量子化ステツプを示す特性曲線図である。

【図１９】従来の動画符号化データの記録順序の説明に供する略線図である。

【符号の説明】

２１……動画符号化／復号化装置、２１Ａ……動画符号化装置、２１Ｂ……動画復号化装置、２５……動き補償回路、２６……動き補償制御ユニツト、２７……予測前フレームメモリ、２８……画像データ符号化回路、２９……変換符号化回路、３０……フレーム間／フレーム内符号化制御ユニツト、３１……フイルタ制御ユニツト、３２……伝送バツフアメモリ、３４……伝送ブロツク設定回路、３５……スレシヨルド制御ユニツト、３６……量子化制御ユニツト、３７……量子化回路、３８……可変長符号可回路。

【図1】

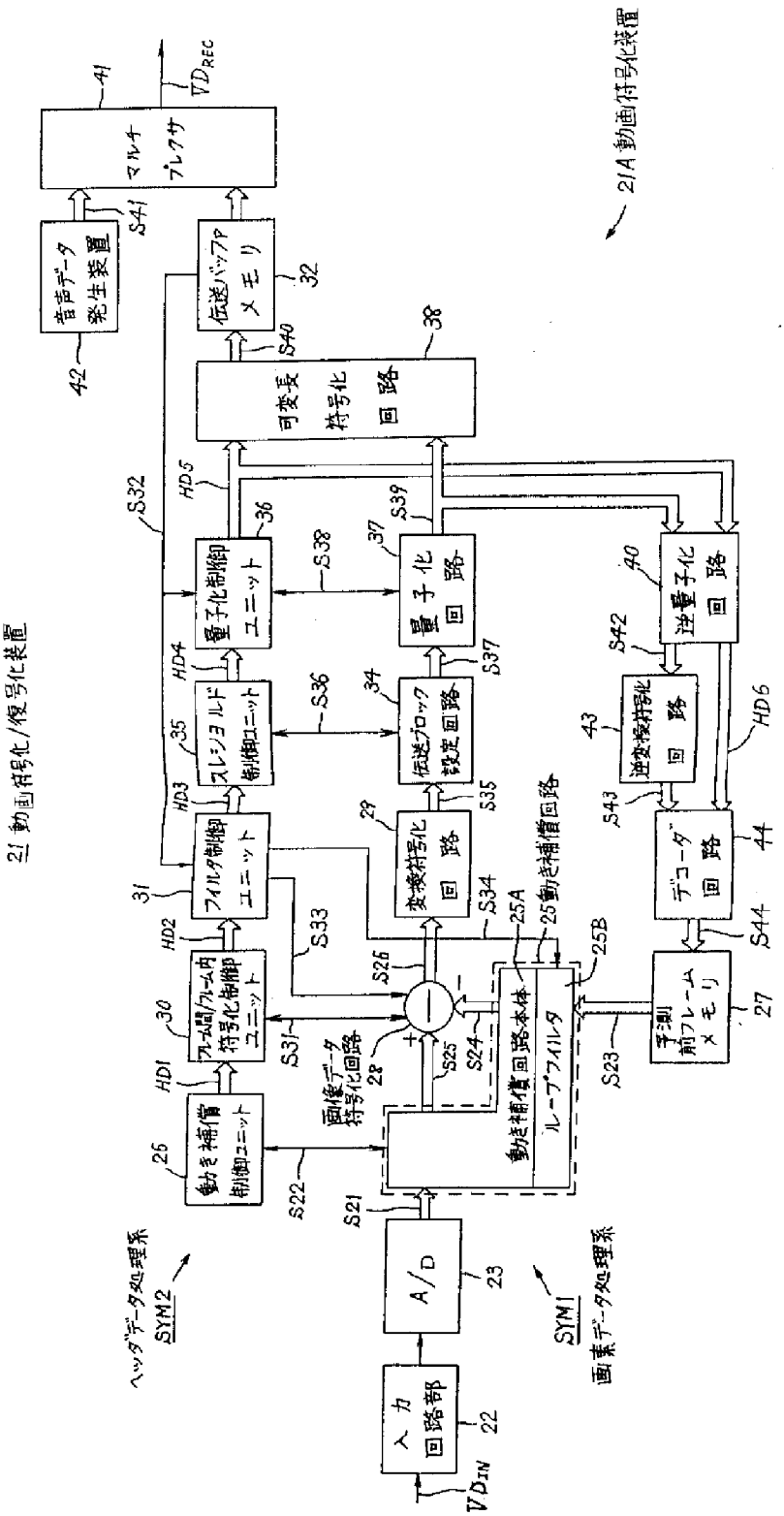


図1　動画符号化装置の構成

【図2】

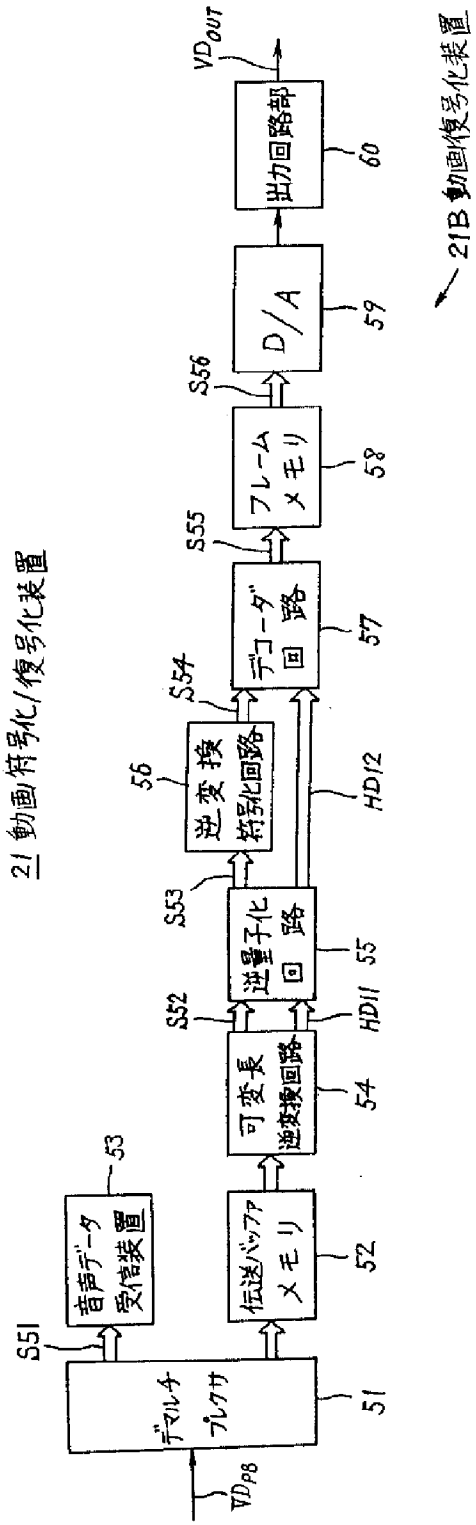


図2　動画復号装置の構成

【図3】

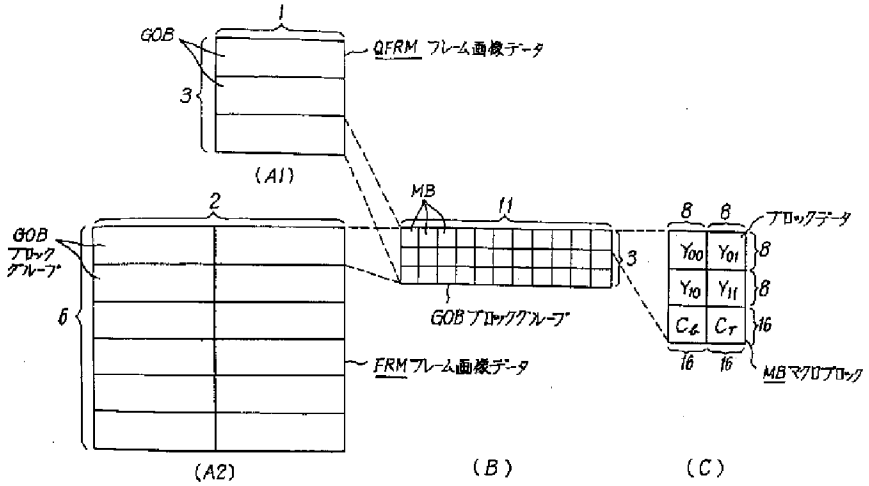


図3　マクロブロックデータの構成

【図6】

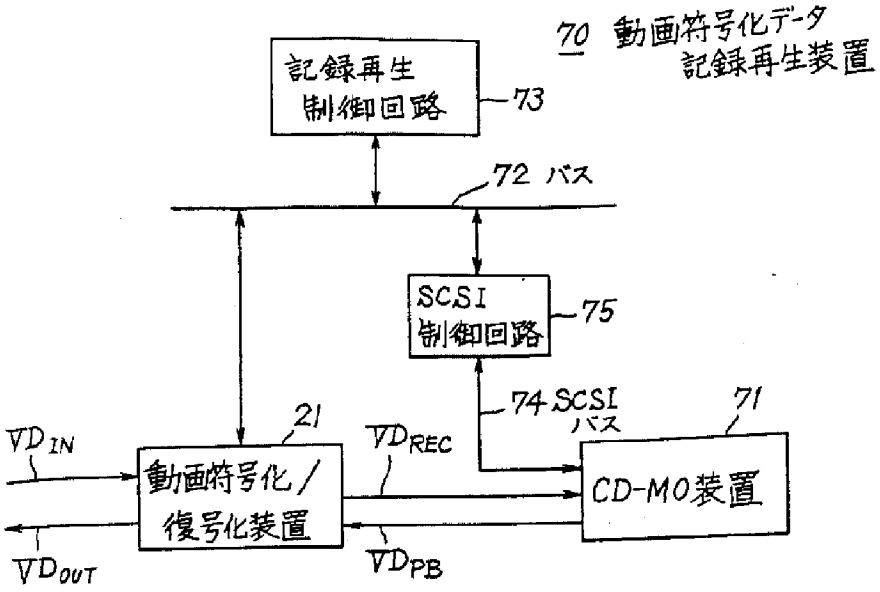


図6　実施例の動画符号化データ記録再生装置の全体構成

【図4】

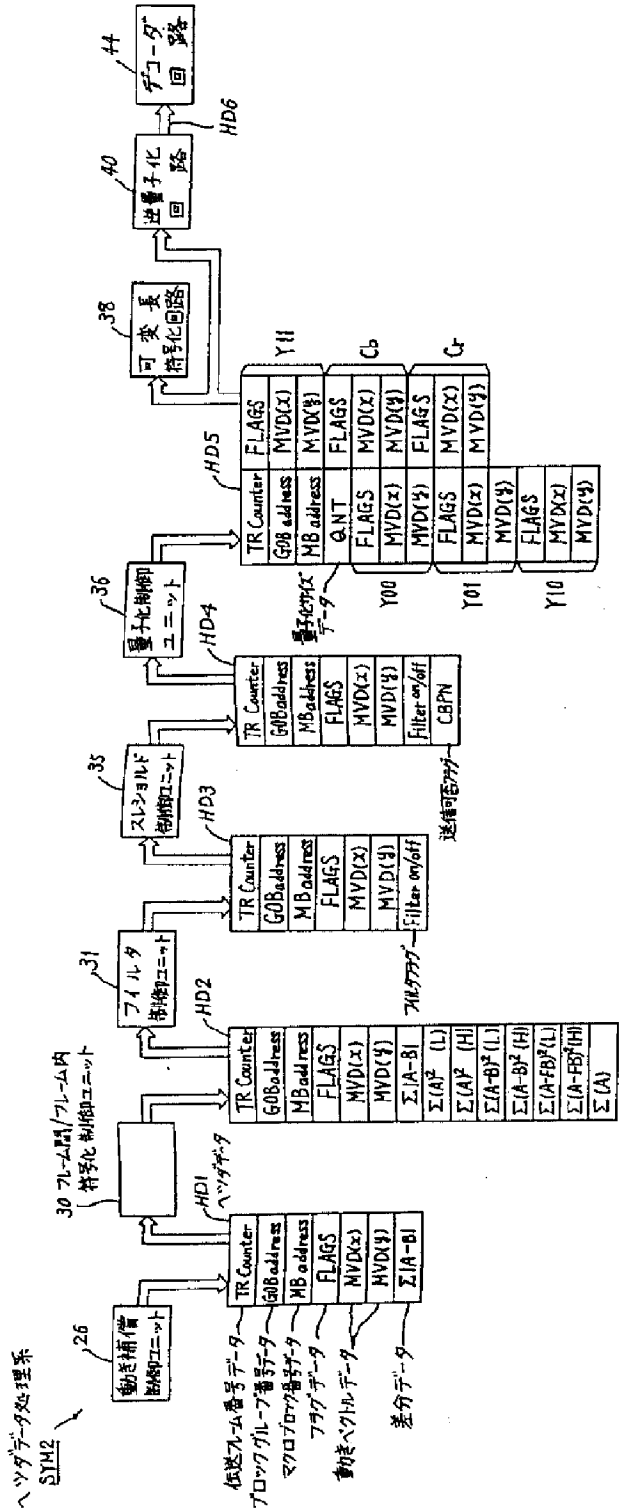


図4　ヘッダデータ処理系の詳細

【図5】

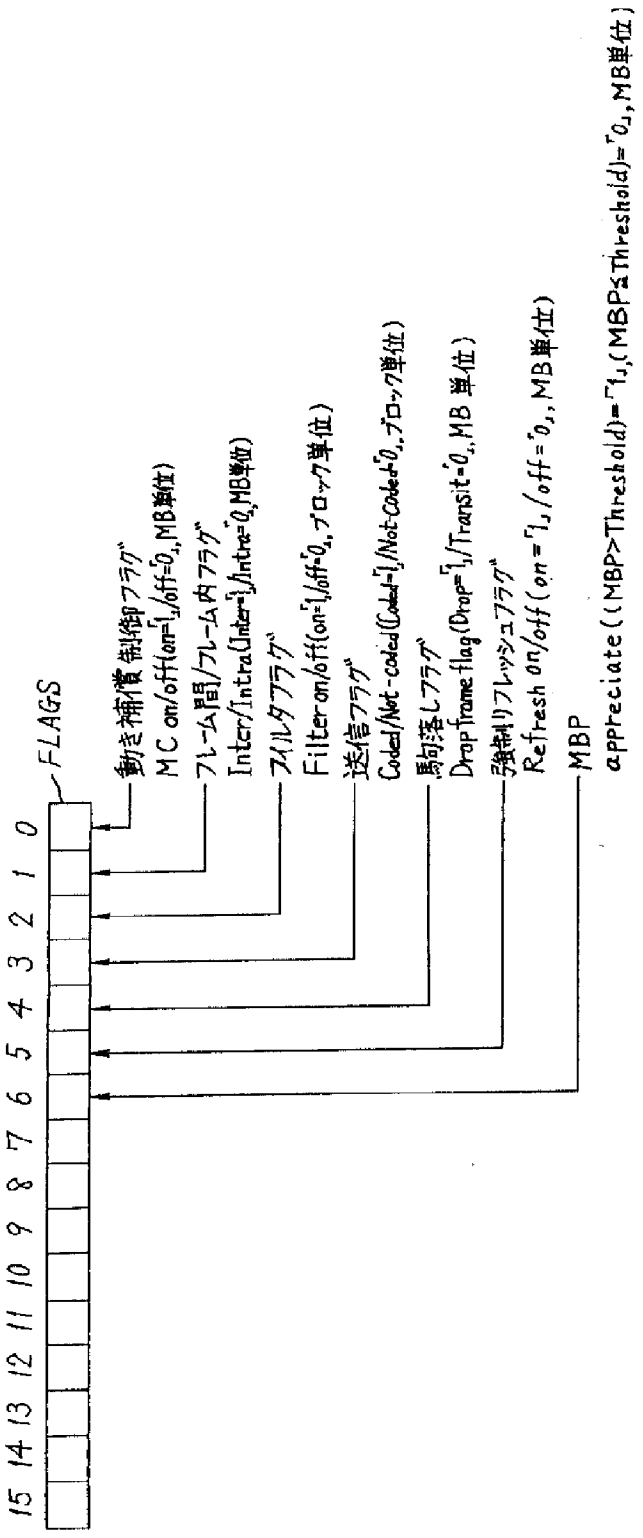


図5　フラグデータの構成

【図11】

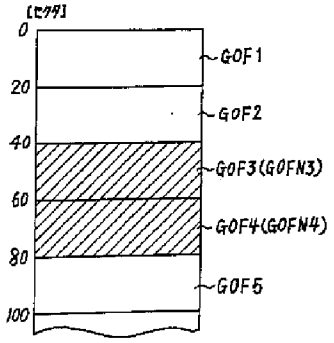


図11　CD－MOディスクの記録領域

【図8】

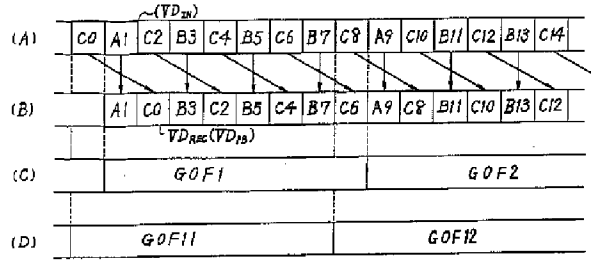


図8　実施例の記録順序

【図7】

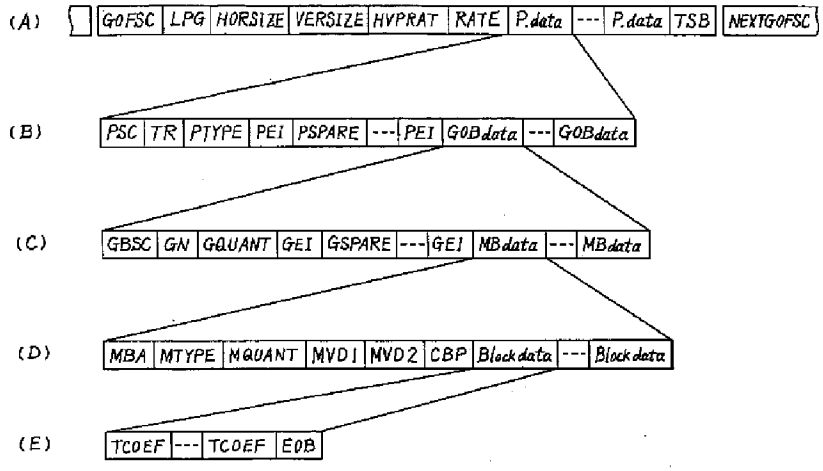


図7　記録再生データのフォーマット

【図12】

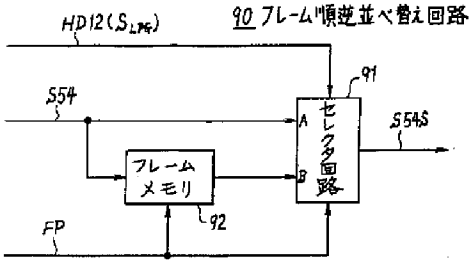


図12　フレーム順逆並べ替え回路

【図13】

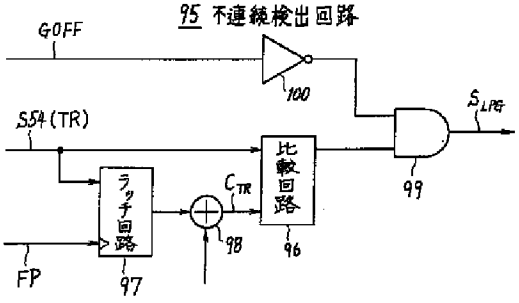


図13　他の実施例

【図9】

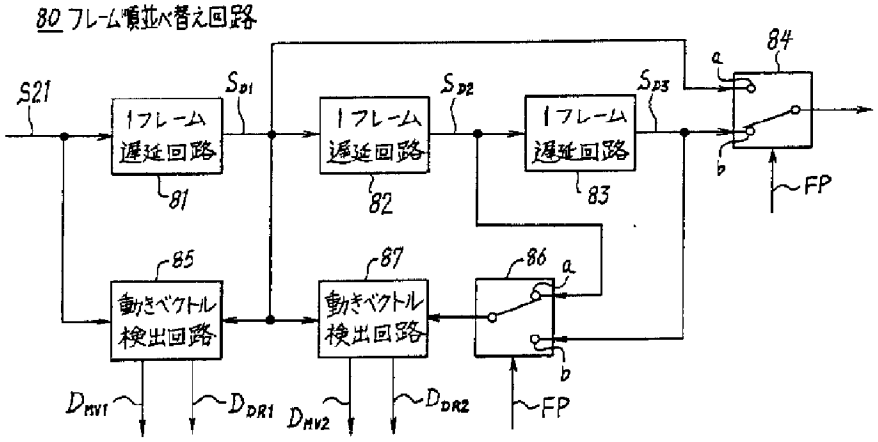


図9　フレーム順並べ替え回路

【図14】

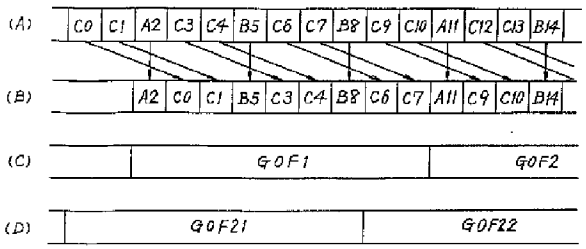


図14　他の実施例の記録順序

【図15】

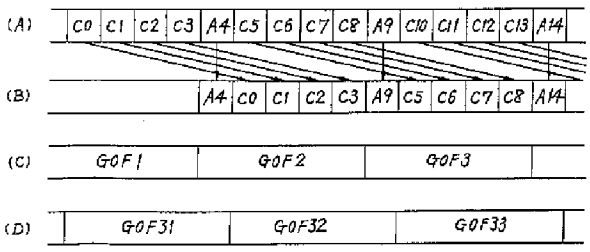


図15　他の実施例の記録順序

【図16】

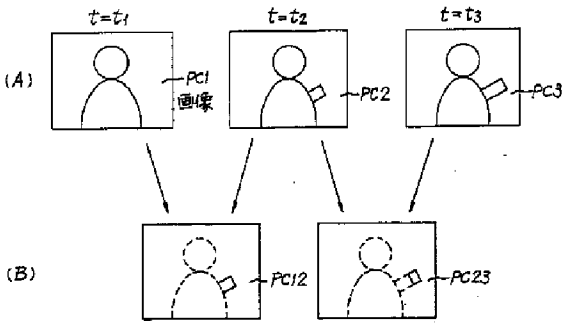


図16　フレーム内／フレーム間符号化処理

【図18】

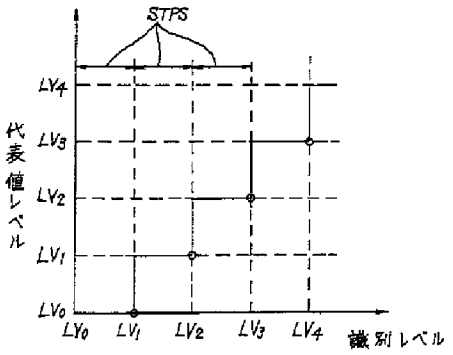


図18　量子化ステップ

【図10】

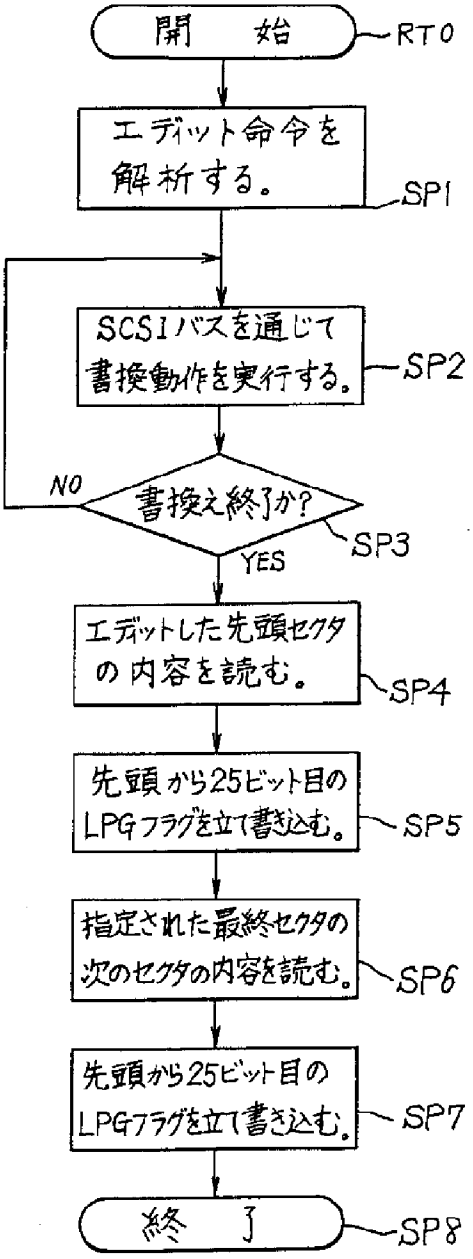


図１０　エデイツト処理手順

【図17】

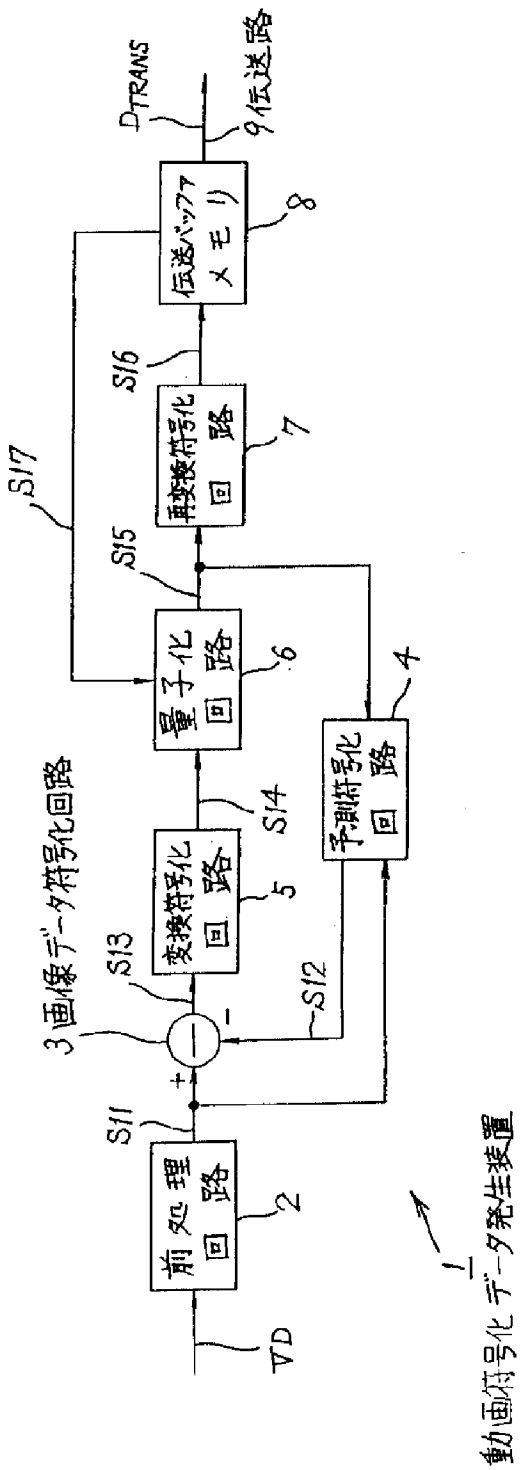


図１７　動画符号化データ発生装置の概略構成

【図１９】

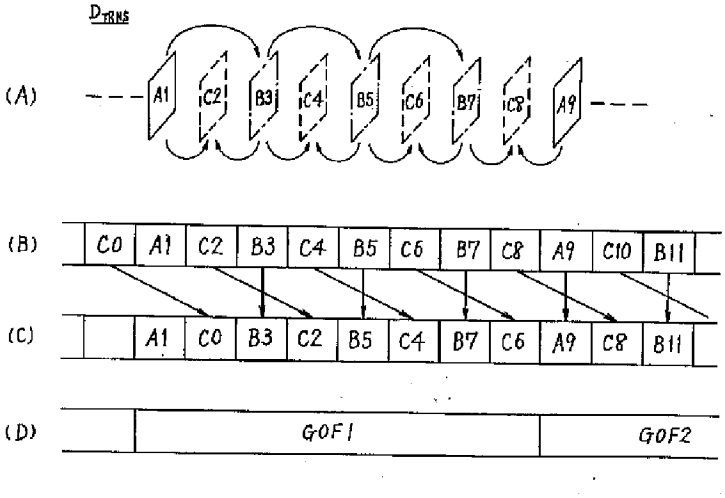


図１９　従来の動画符号化データ記録順序

特許査定

（分割）

特許出願の番号	特願２００６－１２０２９６
起案日	平成２３年　７月２８日
特許庁審査官	畑中　高行　９４６８　５Ｃ００
発明の名称	ビデオ符号化における画像フレームのグループ化
請求項の数	２２
特許出願人	ノキア　コーポレイション
代理人	青木　篤（外　４名）

この出願については、拒絶の理由を発見しないから、特許査定をします。

上記はファイルに記録されている事項と相違ないことを認証する。
認証日　平成23年 7月29日　経済産業事務官　清野　貴明

注意：この書面を受け取った日から３０日以内に特許料の納付が必要です。

１．出願種別　　分割
２．参考文献　　有
３．特許法第３０条適用　　無
４．発明の名称の変更　　無

５．国際特許分類（ＩＰＣ）

Ｈ０４Ｎ　７／１３　Ｚ，
Ｈ０３Ｍ　７／３０　Ｚ

６．菌寄託

７．出願日の遡及を認めない旨の表示

参考情報

特許出願の番号　　特願２００６－１２０２９６

１．調査した分野（ＩＰＣ，ＤＢ名）

Ｈ０４Ｎ５／７６－５／７６５
Ｈ０４Ｎ５／８０－５／９５６
Ｈ０４Ｎ７／１０－７／１７３
Ｈ０４Ｎ７／２０－７／６８
Ｈ０４Ｌ１／００
Ｈ０４Ｌ１／０８－１／２４
Ｈ０３Ｍ３／００－１１／００

２．参考特許文献

特開平０４－３４８６９０	（ＪＰ，Ａ）
特開平１１－１７７９８７	（ＪＰ，Ａ）
特表２００５－５１６４９８	（ＪＰ，Ａ）
特開平０８－２６５６９４	（ＪＰ，Ａ）

３．参考図書雑誌

Electronic Patent Application Fee Transmittal

Application Number:	11338996
Filing Date:	25-Jan-2006
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Filer:	Erin M. Nichols
Attorney Docket Number:	TPO.004.C1

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
		Total in USD ($)		**180**

Electronic Acknowledgement Receipt

EFS ID:	10747071
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	10888
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	19-AUG-2011
Filing Date:	25-JAN-2006
Time Stamp:	14:22:14
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$180
RAM confirmation Number	530
Deposit Account	503581
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	IDS-15-TPO004C1.pdf	55995 b5442ce76caeecba58800d3a77ddb39fa117026a	no	2
Warnings:					
Information:					
2	Information Disclosure Statement (IDS) Form (SB08)	1449-15-TPO004C1.pdf	79563 f3a0ccc0f51a4c170497b030a7d644dac48654e8	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Non Patent Literature	FH-TPO004C2.pdf	8756353 38a70434324c1cc76aabfcf8775dd62a5e900674	no	246
Warnings:					
Information:					
4	Foreign Reference	JP8265694.pdf	1967730 d9d9ae848cced278ebeafff485c8fa8376e1300c	no	22
Warnings:					
Information:					
5	Non Patent Literature	JPAllowance.pdf	51955 b2d041bac2aacea9ef9b9171477eb66f75eb9aea	no	3
Warnings:					
Information:					
6	Fee Worksheet (SB06)	fee-info.pdf	30217 146245a5843f74e9359dfc8d62d1c47c5a852da2	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	10941813		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
UNDER 37 C.F.R. § 1.97(c)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant respectfully submits the items of information on the enclosed Form 1449 for the attention of the Examiner in the above-identified application.

This statement should be considered because it is submitted after the mailing date of the first Office action on the merits but before the mailing date of a final Office action under 37 C.F.R. § 1.113 or a Notice of Allowance under 37 C.F.R. § 1.311. Please charge Deposit Account 50-3581 (TPO.004.C1) the amount of $180.00 under 37 C.F.R. § 1.17(p) for consideration of the items.

In accordance with 37 C.F.R. § 1.98(a)(2), only a copy of each foreign document or non-U.S. patent/application listed on the enclosed Form 1449 is provided.

Please note that any notations or markings on the attached documents do not reflect particular relevance, or lack thereof, to the present application, nor were they necessarily made by anyone affiliated with the prosecution of the present application.

No representation is made that a reference is "prior art" within the meaning of 35 U.S.C. §§ 102 and 103. In addition, Applicant does not represent that a reference has been thoroughly reviewed or that any relevance of any portion of a reference is intended, and reserves the right to establish otherwise under 37 C.F.R. § 1.131 or others.

Consideration of the items listed is respectfully requested. According to MPEP § 609, Applicant requests that the Examiner return a copy of the attached Form 1449, marked as being considered and initialed by the Examiner, to the undersigned with the next official communication.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.232.6907

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti
Reg. No. 57,125

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on November 7, 2011.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action dated August 9, 2011, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.111.

Claims 1, 3-10, 12-18, and 20-25 remain pending in the application. Claims 1, 10, and 18 have been amended. No claims have been added, and claims 2, 11, and 19 have been canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 2 of this paper.

REMARKS/ARGUMENTS begin on page 8.

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions of claims in the application.

1. (Currently amended) A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising:

decoding from the compressed video sequence an indication ~~related to~~informing an intentional discontinuity of numbering of the image frames;

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and

using the image frames in the buffer memory in the decoding process.

2. (Canceled)

3. (Previously presented) A method according to claim 1, further comprising

using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

4. (Original) A method according to claim 1, wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.

5. (Previously presented) A method according to claim 1, further comprising

continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

6. (Previously presented) A method according to claim 1, further comprising

entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

7. (Previously presented) A method according to claim 6, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.

8. (Previously presented) A method according to claim 1, wherein

configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.

9. (Previously presented) A method according to claim 3, further comprising

detecting an error in the compressed video sequence on a basis of a reference in the decoding process to an image frame corresponding to the at least one discontinuity in the numbering of image frames.

10. (Currently amended) A video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding, the video decoder being configured to

decode from the compressed video sequence an indication ~~related to~~informing an intentional discontinuity of numbering of the image frames;

configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and

use the image frames in the buffer memory in the decoding process.

11. (Canceled)

12. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

use the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

13. (Original) A video decoder according to claim 10, wherein

the number of image frames in the buffer memory are used in sliding window buffering of image frames.

14. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

continue the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

15. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

enter into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

16. (Previously presented) A video decoder according to claim 15, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.

17. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

configure the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

18. (Currently amended) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, image frames of said video sequence being entered into a buffer memory in connection with the decoding, the computer program product comprising:

a computer program code for decoding from the compressed video sequence an indication ~~related to~~informing an intentional discontinuity of numbering of the image frames;

a computer program code for configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and

a computer program code for including the image frames in the buffer memory in the decoding process.

19. (Canceled)

20. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for including the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

21. (Original) A computer program product according to claim 18, the computer program product further comprising

a computer program code for using the number of image frames in the buffer memory in sliding window buffering of image frames.

22. (Previously presented) A computer program product according to claim 18, the computer program product further comprising

a computer program code for continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

23. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

24. (Previously presented) A computer program product according to claim 23, the computer program product further comprising:

a computer program code for designating said filler frames by an identifier indicating that the filler frames do not belong to the compressed video sequence.

25. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

26-31. (Canceled)

REMARKS/ARGUMENTS

The rejections presented in the Office action dated August 9, 2011 (hereinafter "Office action"), have been considered. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

Claims 2, 11, and 19, have been canceled without prejudice or disclaimer rendering the respective objections and rejections of these claims moot. Applicant reserves the opportunity to pursue the subject matter of these claims in continuing applications and requests that the respective objections and rejections be removed and withdrawn.

Applicant respectfully traverses the §§ 102(e) and 103(a) rejections based on the teachings of U.S. Patent No. 7,103,669 to Apostolopoulos (hereinafter "Apostolopoulos"); however, in the interests of advancing prosecution and in a *bona fide* attempt to advance the application to allowance, the independent claims have been amended to clarify particular aspects of the claimed invention. For example, the independent claims have been amended to characterize that the decoded indication informs an intentional discontinuity of numbering of the image frames. Support for these changes may be found in the original specification, for example, in original (and previously allowed) claims 2, 11, and 19; and at paragraph [0017]; therefore, the changes do not introduce new matter. Since neither Apostolopoulos nor Langberg teaches or suggests at least an indication that informs an intentional discontinuity of numbering, as now claimed in the independent claims, the rejections would be improper. Applicant accordingly requests that the rejections be withdrawn.

As set forth above, neither of the asserted references teaches or suggests at least decoding an indication that informs an intentional discontinuity of numbering of the image frames. Again, Apostolopoulos fails to teach or suggest each of the now-claimed elements of the independent claims since the asserted indication of Apostolopoulos addresses only discontinuities of numbering caused by errors. In view of these changes, elements now included in each of the pending claims are not found in Apostolopoulos, either alone or as modified, therefore, each of the prior art rejections would be improper. Applicant accordingly requests that each of the § 102(e) and § 103(a) rejections be withdrawn.

To the extent that the current response does not respond to any characterization in the Office action of the asserted art or of the claimed subject matter, or to any application in the Office action of the asserted art to any claimed subject matter, it is stated for the record that any such lack of response should not be interpreted as an acquiescence to such characterizations or applications. A detailed discussion of each of the Office action's characterizations, or any other assertions or statements beyond that provided above is unnecessary in view of the present response. The right to address in detail any such assertions or statements in the future is reserved.

Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C1) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the undersigned attorney of record invites the Examiner to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.232.6907

Date: November 7, 2011

By: /Erin Nichols Matkaiti/

Erin Nichols Matkaiti
Reg. No. 57,125

Electronic Acknowledgement Receipt

EFS ID:	11351147
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	10888
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	07-NOV-2011
Filing Date:	25-JAN-2006
Time Stamp:	15:56:44
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		OAR-TPO004C1-OA09082011.pdf	86461 c5b037f9765d6c128b393bf9b372b7e00d80ffad	yes	9

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment/Req. Reconsideration-After Non-Final Reject	1	1
	Claims	2	7
	Applicant Arguments/Remarks Made in an Amendment	8	9
Warnings:			
Information:			
	Total Files Size (in bytes):	86461	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
11/338,996	01/25/2006	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY ☐ RATE ($)	SMALL ENTITY ☐ FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT 11/07/2011	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 22	Minus	** 31	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	* 3	Minus	***6	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

AMENDMENT	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/KATRINA . TURNER/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,996	01/25/2006	Miska Hannuksela	TPO.004.C1	5361

10888 7590 01/09/2012

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

NOTIFICATION DATE	DELIVERY MODE
01/09/2012	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

tdotter@hfiplaw.com
roswood@hfiplaw.com

Office Action Summary	Application No.	Applicant(s)	
	11/338,996	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	SIU LEE	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 November 2011*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *1,3-10,12-18 and 20-25* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1,3-7,9,10,12-16,18 and 20-24* is/are rejected.

8)☒ Claim(s) *8,17 and 25* is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☒ The drawing(s) filed on *25 January 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

12)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

13)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☒ Certified copies of the priority documents have been received in Application No. 10/306,942.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments with respect to claims 1-27 have been considered but are moot in view of the new ground(s) of rejection because of the amendment.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1, 3, 5, 9-10, 12, and 14 are rejected under 35 U.S.C. 102(e) as being anticipated by Apostolopoulos (US 7,103,669 B2).

(1) Regarding claim 1:

Apostolopoulos disclose a method for decoding a compressed video sequence (video sequence 354 in figure 5), wherein image frames are entered into a buffer memory in connection with the decoding (frame memory for storing the decoded frames from decode 320 of the video sequence 356 as shown in figure 5), the method comprising:

decoding from the compressed video sequence an indication (frame number of the video sequence 354) related to an intentional discontinuity of numbering of the image frame (due to the diversity transmission of the video sequence as shown in figure 3 (2 paths, one for odd frame and one for even frame) or in figure 1 (3 paths), the frame number of the video sequences 356 and 354 in figure 5 indicate an intentional discontinuity numbering of the image frame as shown in figure 4);

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames (frame memory for video sequence 356 provide diversity frame to the state recovery 526 for recovery of one of lost video frame in the video sequence 354, column 11, lines 4-52); and

using the image frames in the buffer memory in the decoding process (the examiner interprets the process within the dot line as shown in figure 9 as the decoding process, as the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well as P1 and P3, column 11, lines 14-18, therefore, the usage of even frame in the recovery of the odd frame as used in the decoding process).

(2) Regarding claim 10:

Apostolopoulos discloses a video decoder (multiple state video decoding 128 in figure 1 comprises decode block 320 and 322 as shown in figure 5) for decoding a compressed video sequence, the video decoder being connected to a buffer memory

(frame memory as shown in figure 5) wherein image frames are entered in connection with the decoding, the video decoder being configured to

decode from the compressed video sequence (video sequence 354 in figure 5) an indication related to an intentional discontinuity of numbering of the image frames (due to the diversity transmission of the video sequence as shown in figure 3 (2 paths, one for odd frame and one for even frame) or in figure 1 (3 paths), the frame number of the video sequences 356 and 354 in figure 5 indicate an intentional discontinuity numbering of the image frame as shown in figure 4); and

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames (frame memory for video sequence 356 provide diversity frame to the state recovery 526 for recovery of one of lost video frame in the video sequence 354, column 11, lines 4-52); and

using the image frames in the buffer memory in the decoding process (the examiner interprets the process within the dot line as shown in figure 9 as the decoding process, as the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18, therefore, therefore, the usage of even frame in the recovery of the odd frame as used in the decoding process).

(3) Regarding claims 3 and 12:

Apostolopoulos further disclose using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames (the video sequence 356 comprises image frame that is not in the video sequence 354 and the merging of the video frames in sequence 356 into the missing frames in the video sequence 354 to reconstruct the video 344 as shown in figure 4, column 9, lines 23-26).

(4) Regarding claim 9:

Apostolopoulos further disclose detecting an error in the compressed video sequence on a basis of a reference in the decoding process to an image frame corresponding to the at least one discontinuity in the numbering of image frames (detection of the error occur in the transmission of the encoded odd frame corresponds to the even frame sequence since the odd frame is missing in the even frame sequence, column 9, lines 66-67).

(5) Regarding claims 5 and 14:

Apostolopoulos further disclose continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames (the receiver 332 includes a state recovery 526 for recovering the lost P-frame P5 of the stream 354 using the information provided by the stream 356, the recovery of P5 enables the decode block 322 to continue with its prediction loop by correctly updating its state information and reconstruct p7 to P13,

column 10, lines 55-60; as the frames are correctly decode by the decoder, frames are enter into the frame memory as a previous decoded frames as shown in figure 5).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 6-7, 15-16, 18, 20-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2) in view of Langberg (US 5,852,630).

(1) Regarding claim 18:

Apostolopoulos disclose a method for decoding a compressed video sequence (video sequence 354 in figure 5), wherein image frames are entered into a buffer memory in connection with the decoding (frame memory for storing the decoded frames from decode 320 of the video sequence 356 as shown in figure 5), the method comprising:

decoding from the compressed video sequence an indication (frame number of the video sequence 354) related to an intentional discontinuity of numbering of the image frame (due to the diversity transmission of the video sequence as shown in figure 3 (2 paths, one for odd frame and one for even frame) or in figure 1 (3 paths), the frame

number of the video sequences 356 and 354 in figure 5 indicate an intentional discontinuity numbering of the image frame as shown in figure 4);

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames (frame memory for video sequence 356 provide diversity frame to the state recovery 526 for recovery of one of lost video frame in the video sequence 354, column 11, lines 4-52); and

using the image frames in the buffer memory in the decoding process (the examiner interprets the process within the dot line as shown in figure 9 as the decoding process, as the state recovery block 526 may recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstruction full frame versions of P2, P4, and P6 through P14, as well we P1 and P3, column 11, lines 14-18, therefore, the usage of even frame in the recovery of the odd frame as used in the decoding process).

Apostolopoulos discloses all of the subject matter as described above except for the method is perform by a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device.

However, Langberg et al. teaches that the method and apparatus for a transceiver warm start activation procedure with precoding can be implemented in software stored in a computer-readable medium. The computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can be contain or store a computer program for use by or in connection with a computer-related system or

method (column 3, lines 51-65). One skilled in the art would have clearly recognized that the method of Apostolopoulos would have been implemented in a software. The implemented software would perform same function of the hardware for less expense, adaptability, and flexibility. Therefore, it would have been obvious to have used the software as taught by Langberg for implementing the method of Apostolopoulos in order to reduce cost and improve the adaptability and flexibility of the communication system.

(2) Regarding claim 20:

Apostolopoulos further disclose using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames (the video sequence 356 comprises image frame that is not in the video sequence 354 and the merging of the video frames in sequence 356 into the missing frames in the video sequence 354 to reconstruct the video 344 as shown in figure 4, column 9, lines 23-26).

(3) Regarding claim 21:

Apostolopoulos disclose all subject matter as discussed in claim 1 but fails to explicitly disclose wherein the number of image frames in the buffer memory is used in sliding widow buffering of image frames.

However, as shown in figure 5, decoded frame as input into the frame memory as previous decode frames, these frames are used for recovery of a lost frame (recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstructed full frame versions of P2, P4, P6 through P14 as well as P1 and P3, column 11, lines 14-18), therefore, the frame

memory have to store the decoded frame P2 to P14 in the frame memory on the even frame number sequence and obviously, the frame memory have to store the decoded frame P1 to P13 in the frame memory on the odd frame number sequence, therefore, the examiner interprets the frame memory are used in sliding window buffering of the decode image frame.

It is desirable to have the number of image frames in the buffer memory are used in sliding widow buffering of image frames because the buffer of image frame is need for recovering of a full frame version of a lost frame and provide the benefit of increase the image quality of the recovered video sequence.

(4) Regarding claim 22:

Apostolopoulos further disclose continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames (the receiver 332 includes a state recovery 526 for recovering the lost P-frame P5 of the stream 354 using the information provided by the stream 356, the recovery of P5 enables the decode block 322 to continue with its prediction loop by correctly updating its state information and reconstruct p7 to P13, column 10, lines 55-60; as the frames are correctly decode by the decoder, frames are enter into the frame memory as a previous decoded frames as shown in figure 5).

(5) Regarding claims 6, 15, and 23:

Apostolopoulos discloses all subject matter as discussed in claim 1 except explicitly discloses entering into the buffer memory a number of filter frames corresponding to the at least one discontinuity in the numbering of the image frame.

However, Apostolopoulos discloses the entering of the decoded frames to the frame memory in figure 5 corresponding to the discontinuity in the numbering of the other even or odd frame sequence, these odd or even decoded frames are used to replace error frames in the other video sequence (even or odd) by the method as discloses in column 11, lines 48-52, the lost frame may be estimated by replacing it by another correctly decoded frame, the examiner interprets the correctly decode frame that replace the lost frame as the claimed “filler frame”; therefore, the claimed limitation is met.

(6) Regarding claims 7, 16, and 24:

Apostolopoulos further discloses wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence (when using the correctly decoded frames of the other sequence (e.g. Even frame sequence) to replace an odd error frame, the even frame number (identifier) of the even frame sequence indicates that the even frames do not belong to the odd frame sequence as shown in figure 4).

6. Claims 4 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos (US 7,103,669 B2).

Apostolopoulos disclose all subject matter as discussed in claim 1 but fails to explicitly disclose wherein the number of image frames in the buffer memory is used in sliding widow buffering of image frames.

However, as shown in figure 5, decoded frame as input into the frame memory as previous decode frames, these frames are used for recovery of a lost frame (recover the full frame version of P5 by interpolating or estimating the full frame version of P5 by employing any combination of the reconstructed full frame versions of P2, P4, P6 through P14 as well as P1 and P3, column 11, lines 14-18), therefore, the frame memory have to store the decoded frame P2 to P14 in the frame memory on the even frame number sequence and obviously, the frame memory have to store the decoded frame P1 to P13 in the frame memory on the odd frame number sequence, therefore, the examiner interprets the frame memory are used in sliding window buffering of the decode image frame.

It is desirable to have the number of image frames in the buffer memory are used in sliding widow buffering of image frames because the buffer of image frame is need for recovering of a full frame version of a lost frame and provide the benefit of increase the image quality of the recovered video sequence.

Allowable Subject Matter

7. Claims 8, 17 and 25 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 6:15-3:45 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/
Primary Examiner, Art Unit 2611
12/25/2011

Notice of References Cited	Application/Control No. 11/338,996	Applicant(s)/Patent Under Reexamination HANNUKSELA, MISKA	
	Examiner SIU LEE	Art Unit 2611	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-7,103,669	09-2006	Apostolopoulos, John G.	709/231
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 5361

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/338,996	01/25/2006 RULE	375	2611	TPO.004.C1

APPLICANTS
Miska Hannuksela, Tampere, FINLAND;

**** CONTINUING DATA ***************************
This application is a CON of 10/306,942 11/29/2002 PAT 7,894,521

**** FOREIGN APPLICATIONS ***************************
FINLAND 20020127 01/23/2002

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****
02/28/2006

Foreign Priority claimed ☑ Yes ☐ No 35 USC 119(a-d) conditions met ☑ Yes ☐ No Verified and Acknowledged /SIU M LEE/ Examiner's Signature	☐ Met after Allowance Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		FINLAND	4	31	6

ADDRESS
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

TITLE
Grouping of image frames in video coding

FILING FEE RECEIVED	FEES: Authority has been given in Paper	☐ All Fees
2150	No._____ to charge/credit DEPOSIT ACCOUNT	☐ 1.16 Fees (Filing)
	No._____ for following:	☐ 1.17 Fees (Processing Ext. of time)
		☐ 1.18 Fees (Issue)
		☐ Other _____
		☐ Credit

FORM 1449* SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT IN AN APPLICATION (Use several sheets if necessary)	Docket Number: TPO.004.C1	Application Number: 11/338,996
	Applicant: HANNUKSELA	
	Filing Date: 01/25/2006	Group Art Unit: 2611

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO
/SL/	JP 8265694	10/11/1996	JP			Abstract	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/SL/	08/02/2011	Notice of Allowance dated 8/2/2011 from Japanese Application No. 2006-120296, 3 pages.
/SL/	08/19/2011	File History for U.S. Serial No.11/338,934 as retrieved from the U.S. Patent and Trademark Office PAIR System on 8/19/2011, 246 pages.

EXAMINER /Siu Lee/	DATE CONSIDERED 12/25/2011

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; draw line through citation if not in conformance and not considered. Include copy of this form for next communication to the Applicant.

*Substitute Disclosure Statement Form (PTO-1449)

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	06/13/2010	11/01/2010	02/14/2011	07/26/2011	12/25/2011				
	1	✓	✓	✓	✓	✓				
	2	✓	✓	O	✓	-				
	3	✓	✓	✓	✓	✓				
	4	✓	✓	✓	✓	✓				
	5	✓	✓	✓	✓	✓				
	6	✓	✓	✓	✓	✓				
	7	O	O	✓	✓	✓				
	8	O	O	✓	✓	O				
	9	O	✓	✓	✓	✓				
	10	✓	✓	✓	✓	✓				
	11	✓	✓	O	✓	-				
	12	✓	✓	✓	✓	✓				
	13	✓	✓	✓	✓	✓				
	14	✓	✓	✓	✓	✓				
	15	✓	✓	✓	✓	✓				
	16	✓	O	✓	✓	✓				
	17	✓	O	✓	✓	O				
	18	✓	✓	✓	✓	✓				
	19	✓	✓	O	✓	-				
	20	✓	✓	✓	✓	✓				
	21	✓	✓	✓	✓	✓				
	22	✓	✓	✓	✓	✓				
	23	✓	✓	✓	✓	✓				
	24	✓	O	✓	✓	✓				
	25	✓	O	✓	✓	O				
	26	✓	-	-	-	-				
	27	✓	-	-	-	-				
	28	✓	-	-	-	-				
	29	✓	-	-	-	-				
	30	✓	-	-	-	-				
	31	✓	-	-	-	-				

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	18667	video compression	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/24 18:39
S2	2714	S1 and ((tag or number or indicat$3 or frame) with (omit$4 or discontinu$4 or skip$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/24 18:41
S3	1368	S2 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/24 18:41
S4	86	S3 and ((reconstruct$3) with (omit$4 or discontinu$4 or skip$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/24 18:42
S5	2	"5680322".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/26 14:16
S6	2	"5786858".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/26 14:37
S7	2	"6483875".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/07/26 14:38
S8	0	decod$3 with video with indicat$3 with number with (miss$3 or discontinuity or skip$4 or omit$4) frame	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/12/06 13:44
S9	34	decod$3 with video with indicat$3 with number with (miss$3 or discontinuity or skip$4 or omit$4) with frame	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/12/06 13:44

EAST Search History (Interference)

< This search history is empty>

12/25/2011 9:28:47 PM
C:\Users\slee7\Documents\EAST\Workspaces\11338996 201107.wsp

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

SEARCHED

Class	Subclass	Date	Examiner
375	240.27, 240.25	6/14/2010	Siu M. Lee

SEARCH NOTES

Search Notes	Date	Examiner
EAST	6/14/2010	Siu M. Lee
Discuss the application with Chieh Fan	6/14/2010	Siu M. Lee
EAST	11/1/2010	Siu M. Lee
Consult John Peng on the 101 problem in claim 10-17	11/1/2010	Siu M. Lee
Discuss the amendment with Cheih Fan	10/27/2010	Siu M. Lee
EAST	2/14/2011	Siu M. Lee
EAST	7/25/2011	Siu M. Lee
Discuss the amendment with Chieh Fan	7/25/2011	Siu M. Lee
EAST	12/25/2011	Siu M. Lee

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

To: tdotter@hfiplaw.com,roswood@hfiplaw.com,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 10888

Jan 09, 2012 05:22:32 AM

Dear PAIR Customer:

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 10888 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
11338996	CTFR	01/09/2012	TPO.004.C1
	892	01/09/2012	TPO.004.C1
	1449	01/09/2012	TPO.004.C1

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on March 5, 2012.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

RESPONSE UNDER 37 C.F.R. § 1.116

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action dated January 9, 2012, setting a three (3) month shortened statutory period for response, Applicant submits the present response under 37 C.F.R. § 1.116.

Claims 1, 3-10, 12-18, and 20-25 remain pending in the application. No claims have been amended, added, or canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

THE CLAIMS are reflected in the listing of claims which begins on page 2.

REMARKS/ARGUMENTS begin on page 8.

THE CLAIMS

While no amendments, additions, or cancellations of claims are effected via this paper, this listing of claims is provided for the convenience of the Examiner.

1. (Previously presented) A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising:

decoding from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames;

configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and

using the image frames in the buffer memory in the decoding process.

2. (Canceled)

3. (Previously presented) A method according to claim 1, further comprising

using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

4. (Original) A method according to claim 1, wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.

5. (Previously presented) A method according to claim 1, further comprising

continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

6. (Previously presented) A method according to claim 1, further comprising

entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

7. (Previously presented) A method according to claim 6, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.

8. (Previously presented) A method according to claim 1, wherein

configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.

9. (Previously presented) A method according to claim 3, further comprising

detecting an error in the compressed video sequence on a basis of a reference in the decoding process to an image frame corresponding to the at least one discontinuity in the numbering of image frames.

10. (Previously presented) A video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding, the video decoder being configured to

decode from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames;

configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and

use the image frames in the buffer memory in the decoding process.

11. (Canceled)

12. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

use the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

13. (Original) A video decoder according to claim 10, wherein

the number of image frames in the buffer memory are used in sliding window buffering of image frames.

14. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

continue the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

15. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

enter into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

16. (Previously presented) A video decoder according to claim 15, wherein

said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.

17. (Previously presented) A video decoder according to claim 10, wherein the video decoder is further configured to

configure the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

18. (Previously presented) A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, image frames of said video sequence being entered into a buffer memory in connection with the decoding, the computer program product comprising:

a computer program code for decoding from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames;

a computer program code for configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and

a computer program code for including the image frames in the buffer memory in the decoding process.

19. (Canceled)

20. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for including the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

21. (Original) A computer program product according to claim 18, the computer program product further comprising

a computer program code for using the number of image frames in the buffer memory in sliding window buffering of image frames.

22. (Previously presented) A computer program product according to claim 18, the computer program product further comprising

a computer program code for continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

23. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.

24. (Previously presented) A computer program product according to claim 23, the computer program product further comprising:

a computer program code for designating said filler frames by an identifier indicating that the filler frames do not belong to the compressed video sequence.

25. (Previously presented) A computer program product according to claim 18, the computer program product further comprising:

a computer program code for configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.

26-31. (Canceled)

REMARKS/ARGUMENTS

The rejections presented in the Office action dated January 9, 2012 (hereinafter "Office action"), have been considered but are believed to be in error. Reconsideration of the pending claims and allowance of the application in view of the present response is respectfully requested.

Applicant appreciates the indication of allowability for claims 8, 17, and 25, but respectfully declines the opportunity to amend the claims at this time.

Applicant respectfully traverses each of the rejections (§§ 102(e) and 103(a)) based on the teachings of U.S. Patent No. 7,103,669 to Apostolopoulos (hereinafter "Apostolopoulos") because Apostolopoulos does not teach or suggest each of the asserted claim elements. For example, Apostolopoulos has not been shown to teach or suggest at least configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames, as claimed in each of the independent claims. While the Examiner alleges that the frame memory provides a diversity frame to the state recovery 526, Apostolopoulos does not mention the term "diversity frame." Also, the frame supplied to the state recovery 526 is used to average, interpolate, or estimate a missing odd frame based on image data from even frames (Col. 11, lines 4-23) – it does not correspond to the discontinuity in the numbering of the image frames, as claimed. Notably, state recovery block 526 is no buffer memory, but is instead an operational block for recovering an odd frame (such as P5) missing from a sequence of even frames. Apostolopoulos is directed to creating a full frame version of the odd frame (*e.g.*, P5) missing between two even frames (*e.g.*, P4 and P6), wherein extensive data processing and calculations are needed. Thus, Apostolopoulos does not teach or suggest a buffer memory, which provides a number of image frames corresponding to a discontinuity in the numbering of the image frames, as claimed. Without correspondence to each of the claimed elements, the rejections are in error.

With particular respect to the § 102(e) rejection, to anticipate a claim the asserted reference must teach every element of the claim. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a

single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The Federal Circuit also recently held that "Because the hallmark of anticipation is prior invention, the prior art reference—in order to anticipate under 35 U.S.C. § 102—must not only disclose all elements of the claim within the four corners of the document, but must also disclose those elements 'arranged as in the claim.'" (*Net Moneyin, Inc. v. Verisign, Inc.*, 545 F.3d 1359, 2008 WL 4614511 (Fed. Cir. 2008) quoting *Connell v. Sears, Roebuck & Co.*, 722 F.2d 1542, 1548 (Fed. Cir. 1983)). Therefore, all claim elements, must be found in the prior art reference to maintain a rejection based on 35 U.S.C. § 102. Applicant respectfully submits that Apostolopoulos does not teach every element of independent claims 1, 10, and 18, in the requisite detail, and therefore fails to anticipate claims 1, 3, 5, 9, 10, 12, and 14. Applicant accordingly requests that the rejection be withdrawn.

By similar reasoning, the claims which depend on claims 1, 10, or 18, are also not anticipated by, or obvious in view of, Apostolopoulos, whether the reference is considered alone or modified as asserted. Applicant respectfully requests withdrawal of the rejection of claims 1, 3-10, 12-18, and 20-25, and submits that all claims are in condition for allowance.

It should be noted that Applicant does not acquiesce to the Examiner's statements or conclusions concerning what would have been obvious to one of ordinary skill in the art, obvious design choices, common knowledge at the time of Applicant's invention, officially noticed facts, and the like. Applicant reserves the right to address in detail the Examiner's characterizations, conclusions, and rejections in future prosecution.

Authorization is given to charge Deposit Account No. 50-3581 (TPO.004.C1) any necessary fees for this filing. If the Examiner believes it necessary or helpful, the undersigned attorney of record invites the Examiner to contact the undersigned attorney to discuss any issues related to this case.

Respectfully submitted,

HOLLINGSWORTH & FUNK, LLC
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
952.232.6907

Date: March 5, 2012

By: /Erin Nichols Matkaiti/

Erin Nichols Matkaiti
Reg. No. 57,125

Electronic Acknowledgement Receipt

EFS ID:	12222605
Application Number:	11338996
International Application Number:	
Confirmation Number:	5361
Title of Invention:	Grouping of image frames in video coding
First Named Inventor/Applicant Name:	Miska Hannuksela
Customer Number:	10888
Filer:	Erin M. Nichols
Filer Authorized By:	
Attorney Docket Number:	TPO.004.C1
Receipt Date:	05-MAR-2012
Filing Date:	25-JAN-2006
Time Stamp:	13:40:36
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		OAR-TPO004C1-OA09012012.pdf	97139 58f86bcd23a09572762d3ea808dc4868e51eae65	yes	10

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment After Final	1	1
	Claims	2	7
	Applicant Arguments/Remarks Made in an Amendment	8	10
Warnings:			
Information:			
	Total Files Size (in bytes):	97139	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☒ To be Mailed
11/338,996	01/25/2006	

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY ☐ RATE ($)	SMALL ENTITY ☐ FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
FOR	NUMBER FILED	NUMBER EXTRA					
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT 03/05/2012

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 22	Minus	** 31	= 0	X $ =		OR	X $60=	0
Independent (37 CFR 1.16(h))	* 3	Minus	***6	= 0	X $ =		OR	X $250=	0
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/NICHELE PETERSON/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OK to entert

/SL/

03/07/2012

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on March 5, 2012.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

RESPONSE UNDER 37 C.F.R. § 1.116

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action dated January 9, 2012, setting a three (3) month shortened statutory period for response, Applicant submits the present response under 37 C.F.R. § 1.116.

Claims 1, 3-10, 12-18, and 20-25 remain pending in the application. No claims have been amended, added, or canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

THE CLAIMS are reflected in the listing of claims which begins on page 2.

REMARKS/ARGUMENTS begin on page 8.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 5361

SERIAL NUMBER	FILING or 371(c) DATE RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/338,996	01/25/2006	375	2611	TPO.004.C1

APPLICANTS
Miska Hannuksela, Tampere, FINLAND;

**** CONTINUING DATA ***************************
This application is a CON of 10/306,942 11/29/2002 PAT 7,894,521

**** FOREIGN APPLICATIONS ***************************
FINLAND 20020127 01/23/2002

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****
02/28/2006

Foreign Priority claimed ☑ Yes ☐ No 35 USC 119(a-d) conditions met ☑ Yes ☐ No Verified and Acknowledged /SIU M LEE/ Examiner's Signature	☐ Met after Allowance Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		FINLAND	4	31	6

ADDRESS
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

TITLE
Grouping of image frames in video coding

FILING FEE RECEIVED		
2150	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

BIB (Rev. 05/07).

Issue Classification

Application/Control No.	Applicant(s)/Patent Under Reexamination
11338996	HANNUKSELA, MISKA
Examiner	**Art Unit**
SIU LEE	2611

ORIGINAL		INTERNATIONAL CLASSIFICATION	
CLASS	SUBCLASS	CLAIMED	NON-CLAIMED
375	240.27	H 0 4 B 1 / 66 (2006.01.01)	

CROSS REFERENCE(S)

CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
375	240.27	240.25			

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☒ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	15	17												
	2	16	18												
2	3		19												
3	4	17	20												
4	5	18	21												
5	6	19	22												
6	7	20	23												
7	8	21	24												
8	9	22	25												
9	10		26												
	11		27												
10	12		28												
11	13		29												
12	14		30												
13	15		31												
14	16														

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	22	
/SIU LEE/ Primary Examiner.Art Unit 2611	03/07/2012	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	1

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

SEARCHED

Class	Subclass	Date	Examiner
375	240.27, 240.25	6/14/2010	Siu M. Lee

SEARCH NOTES

Search Notes	Date	Examiner
EAST	6/14/2010	Siu M. Lee
Discuss the application with Chieh Fan	6/14/2010	Siu M. Lee
EAST	11/1/2010	Siu M. Lee
Consult John Peng on the 101 problem in claim 10-17	11/1/2010	Siu M. Lee
Discuss the amendment with Cheih Fan	10/27/2010	Siu M. Lee
EAST	2/14/2011	Siu M. Lee
EAST	7/25/2011	Siu M. Lee
Discuss the amendment with Chieh Fan	7/25/2011	Siu M. Lee
EAST	12/25/2011	Siu M. Lee
EAST	3/7/2012	Siu M. Lee
Inventor name search - Miska Hannuksela	3/7/2012	Siu M. Lee

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner
375	240.27, 240.25	3/7/2012	Siu M. Lee

OK to enter. /SL/

03/12/2012

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Applicant:	HANNUKSELA	Examiner:	Lee, S.
Serial No.:	11/338,996	Group Art Unit:	2611
Filed:	January 25, 2006	Docket No.:	TPO.004.C1
Confirmation No.:	5361	Customer No.:	10888
Title:	GROUPING OF IMAGE FRAMES IN VIDEO CODING		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this paper is being electronically transmitted by EFS-WEB to the United States Patent and Trademark Office on January 31, 2011.

By: /Erin Nichols Matkaiti/
Erin Nichols Matkaiti

AMENDMENT & RESPONSE UNDER 37 C.F.R. § 1.116

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office action dated December 9, 2010, setting a three (3) month shortened statutory period for response, Applicant submits the present amendment and response under 37 C.F.R. § 1.116.

Claims 1-25 remain pending in the application. Claim 10 has been amended. No claims have been added or canceled. Applicant respectfully requests reconsideration and allowance of all pending claims.

AMENDMENTS TO THE SPECIFICATION begin on page 2.

AMENDMENTS TO THE CLAIMS are reflected in the listing of claims which begins on page 3 of this paper.

REMARKS/ARGUMENTS begin on page 9.

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L8	180	((MISKA) near2 (HANNUKSELA)).INV.	US-PGPUB; USPAT; USOCR	ADJ	ON	2012/03/07 18:41
S1	0	"7103669".pn. and (memory or buffer)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 17:15
S2	1	"7103669".pn. and (stor$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 17:15
S3	11	sliding window buffering	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 17:32
S4	6	S3 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 17:34
S5	0	"7103669".pn. and (frame near2 memory)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 19:35
S6	1	"7103669".pn. and (previous frame)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/13 19:35
S9	22	buffer$3 with slid$3 window with video	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 09:32
S10	6	S9 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 09:32
S11	1	"7103669".pn. and replac$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 09:50
S12	1	11/338996 and intentional	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 09:55
S13	682	us$3 with memory (location or indicat$3) with frame	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:18
S14	2	S13 and frame reconstruction	US-PGPUB; USPAT; USOCR; FPRS; EPO;	ADJ	ON	2011/02/14 10:18

			JPO; DERWENT; IBM_TDB			
S15	64	memory (location or indicat$3) with frame number	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:23
S16	57	S15 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:23
S17	0	S16 and (memory (location or indicat$3) with frame number with instead)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:23
S18	0	S16 and (memory (location or indicat$3) with frame number with reaplc$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:23
S19	1	S16 and (memory (location or indicat$3) with frame number with replac$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/02/14 10:23
S20	34	decod$3 with video with indicat$3 with number with (miss$3 or discontinuity or skip$4 or omit$4) with frame	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/12/06 13:54
S21	25	S20 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	ADJ	ON	2011/12/06 13:54

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	3354	375/240.27,240.25.ccls.	US-PGPUB; USPAT; UPAD	ADJ	ON	2012/03/07 16:19
L2	198	1 and ((identif$4 or indicat$3 or represent$3 or decod$3) with ((miss$3 or skip$4 or omit$4 or discontinu$4) near3 frame))	US-PGPUB; USPAT; UPAD	ADJ	ON	2012/03/07 16:23
L3	180	2 and ((memory or buffer or stor$3) with decod$3)	US-PGPUB; USPAT; UPAD	ADJ	ON	2012/03/07 16:23
L4	93	3 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; UPAD	ADJ	ON	2012/03/07 16:24
L5	31	3 and ((identif$4 or indicat$3 or represent$3 or decod$3) with ((miss$3 or skip$4 or omit$4 or discontinu$4) near3 (number$3 near2 frame)))	US-PGPUB; USPAT; UPAD	ADJ	ON	2012/03/07 17:16
L6	17	5 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; UPAD	ADJ	ON	2012/03/07 17:18

L7	2	12/458687	US-PGPUB; USPAT; UPAD	ADJ	ON	2012/03/07 18:24
L9	3	((MISKA) near2 (HANNUKSELA)).INV. and (discontinu$5 near4 frame)	US-PGPUB; USPAT; UPAD	ADJ	ON	2012/03/07 18:42
S22	3254	375/240.27,240.25.ccls.	US-PGPUB; USPAT; UPAD	ADJ	ON	2011/12/06 13:56
S23	3254	375/240.27,240.25.ccls.	US-PGPUB; USPAT; UPAD	ADJ	ON	2011/12/06 13:57
S24	1466	S23 and (@pd<="20020123" OR @ad<="20020123" OR @rlad<="20020123" OR @prad<="20020123")	US-PGPUB; USPAT; UPAD	ADJ	ON	2011/12/06 13:57
S25	33	S24 and ((identif$4 or indicat$3 or represent$3) with ((miss$3 or skip$4 or omit$4 or discontinu$4) near3 frame))	US-PGPUB; USPAT; UPAD	ADJ	ON	2011/12/06 14:00
S26	3	S24 and ((identif$4 or indicat$3 or represent$3) with number with ((miss$3 or skip$4 or omit$4 or discontinu$4) with (image or picture) frame))	US-PGPUB; USPAT; UPAD	ADJ	ON	2011/12/06 15:26
S27	23	((identif$4 or indicat$3 or represent$3) with number with ((miss$3 or skip$4 or omit$4 or discontinu$4) with (image or picture) frame))	US-PGPUB; USPAT; UPAD	ADJ	ON	2011/12/06 15:26
S28	20	S27 not S25	US-PGPUB; USPAT; UPAD	ADJ	ON	2011/12/06 15:26

3/7/2012 8:35:24 PM
C:\Users\slee7\Documents\EAST\Workspaces\11338996 201102.wsp

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11338996	HANNUKSELA, MISKA
	Examiner	Art Unit
	SIU M LEE	2611

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☒ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	06/13/2010	11/01/2010	02/14/2011	07/26/2011	12/25/2011	03/07/2012			
1	1	✓	✓	✓	✓	✓	=			
	2	✓	✓	O	✓	-	-			
2	3	✓	✓	✓	✓	✓	=			
3	4	✓	✓	✓	✓	✓	=			
4	5	✓	✓	✓	✓	✓	=			
5	6	✓	✓	✓	✓	✓	=			
6	7	O	O	✓	✓	✓	=			
7	8	O	O	✓	✓	O	=			
8	9	O	✓	✓	✓	✓	=			
9	10	✓	✓	✓	✓	✓	=			
	11	✓	✓	O	✓	-	-			
10	12	✓	✓	✓	✓	✓	=			
11	13	✓	✓	✓	✓	✓	=			
12	14	✓	✓	✓	✓	✓	=			
13	15	✓	✓	✓	✓	✓	=			
14	16	✓	O	✓	✓	✓	=			
15	17	✓	O	✓	✓	O	=			
16	18	✓	✓	✓	✓	✓	=			
	19	✓	✓	O	✓	-	-			
17	20	✓	✓	✓	✓	✓	=			
18	21	✓	✓	✓	✓	✓	=			
19	22	✓	✓	✓	✓	✓	=			
20	23	✓	✓	✓	✓	✓	=			
21	24	✓	O	✓	✓	✓	=			
22	25	✓	O	✓	✓	O	=			
	26	✓	-	-	-	-	-			
	27	✓	-	-	-	-	-			
	28	✓	-	-	-	-	-			
	29	✓	-	-	-	-	-			
	30	✓	-	-	-	-	-			
	31	✓	-	-	-	-	-			

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

10888 7590 03/19/2012

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER
LEE, SIU M

ART UNIT	PAPER NUMBER
2611	

DATE MAILED: 03/19/2012

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,996	01/25/2006	Miska Hannuksela	TPO.004.C1	5361

TITLE OF INVENTION: GROUPING OF IMAGE FRAMES IN VIDEO CODING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1740	$300	$0	$2040	06/19/2012

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

10888 7590 03/19/2012
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,996	01/25/2006	Miska Hannuksela	TPO.004.C1	5361

TITLE OF INVENTION: GROUPING OF IMAGE FRAMES IN VIDEO CODING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1740	$300	$0	$2040	06/19/2012

EXAMINER	ART UNIT	CLASS-SUBCLASS
LEE, SIU M	2611	375-240270

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:
- ❑ Issue Fee
- ❑ Publication Fee (No small entity discount permitted)
- ❑ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ❑ A check is enclosed.
- ❑ Payment by credit card. Form PTO-2038 is attached.
- ❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ❑ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ❑ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,996	01/25/2006	Miska Hannuksela	TPO.004.C1	5361

10888 7590 03/19/2012

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

EXAMINER	
LEE, SIU M	

ART UNIT	PAPER NUMBER
2611	

DATE MAILED: 03/19/2012

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 1146 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 1146 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No.	Applicant(s)	
	11/338,996	HANNUKSELA, MISKA	
	Examiner	Art Unit	
	SIU LEE	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *after final amendment filed on 3/5/2012*.

2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3. ☒ The allowed claim(s) is/are *1, 3-10,12-18 and 20-25*.

4. ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☒ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☒ Certified copies of the priority documents have been received in Application No. *10/306,942* .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

6. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/Siu M Lee/
Primary Examiner, Art Unit 2611

DETAILED ACTION

Allowable Subject Matter

1. Claims 1, 3-10, 12-18, 20-25 are allowed.

2. The following is an examiner's statement of reasons for allowance:

The present invention describes a method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising decoding from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames; configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and using the image frames in the buffer memory in the decoding process. The closest prior art, Apostolopoulos (US 7,103,669 B2) discloses a video communication system that employ multiple state encoding and path diversity but fails to disclose decoding from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames; configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames, this distinct features has been added to independent claims 1, 10 and 18, thus rendering claims 1, 3-10, 12-18, and 20-25 allowable.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SIU LEE whose telephone number is (571)270-1083. The examiner can normally be reached on Mon-Fri, 6:15-3:45 with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Siu M Lee/

Primary Examiner, Art Unit 2611
3/7/2012

To: tdotter@hfiplaw.com,roswood@hfiplaw.com,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 10888

Mar 19, 2012 05:22:03 AM

Dear PAIR Customer:

Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 10888 , have new outgoing correspondence. This correspondence is now available for viewing in Private PAIR.

The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

Disclaimer:
The list of documents shown below is provided as a courtesy and is not part of the official file wrapper. The content of the images shown in PAIR is the official record.

Application	Document	Mailroom Date	Attorney Docket No.
11338996	NOA	03/19/2012	TPO.004.C1

To view your correspondence online or update your email addresses, please visit us anytime at https://sportal.uspto.gov/secure/myportal/privatepair.

If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov with 'e-Office Action' on the subject line or call 1-866-217-9197 during the following hours:

Monday - Friday 6:00 a.m. to 12:00 a.m.

Thank you for prompt attention to this notice,

UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION INFORMATION RETRIEVAL SYSTEM

OPAP IAP4 APR 25 2012 PATENT & TRADEMARK OFFICE

Doc Code:

PTO/SB/82 (01-06)
Approved for use through 12/31/2008. OMB 0651-0035
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REVOCATION OF POWER OF ATTORNEY WITH NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS

Application Number	See attached list
Filing Date	
First Named Inventor	
Art Unit	
Examiner Name	
Attorney Docket Number	

I hereby revoke all previous powers of attorney given in the above-identified application:

☐ A Power of Attorney is submitted herewith.

OR

☑ I hereby appoint the practitioners associated with the Customer Number: 10948

☑ Please change the correspondence address for the above-identified application to:

☑ The address associated with Customer Number: 10948

OR

☐ Firm *or* Individual Name					
Address					
City		State		ZIP	
Country					
Telephone		Email			

I am the:

☐ Applicant/Inventor.

☑ Assignee of record of the entire interest. See 37 CFR 3.71
Statement under 37CFR 3.73(b) is enclosed. (Form PTO/SB/96)

SIGNATURE of Applicant or Assignee of Record

Signature			
Name	Thomas Arria		
Date	4/19/12	Telephone	781-219-8766

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☐ *Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.36. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending on the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPAP IAP4
APR 25 2012
PATENT & TRADEMARK OFFICE

Nokia Ref.	H&S Ref.	Filing Date	Application No.	Assignment Reel/Frame
NC52687US	800.0266.U1(US)	10/30/2009	12/598368	024739/0870
NC76407US	800.0785.U1(US)	9/27/2011	13/246246	027696/0511
NC46779US	800.0833.U1(US)	2/24/2009	12/438666	027675/0965
NC17810US	800.0870.U1(US)	6/28/2006	11/478555	018643/0540
NC47719US	800.0871.U1(US)	9/8/2006	11/517884	018274/0732
NC52605US	800.0872.U1(US)	4/24/2007	11/789267	019299/0978
NC53552US	800.0873.U1(US)	12/21/2006	11/614498	018963/0980 019229/0394
NC53989US	800.0874.U1(US)	9/13/2007	11/854985	020157/0796
NC57047US	800.0875.U1(US)	3/19/2008	12/051787	021102/0082
NC60014US	800.0876.U1(US)	5/13/2008	12/120226	021488/0823
NC62647US	800.0877.U1(US)	12/18/2007	11/958967	021184/0418 023346/0810
NC70507US	800.0878.U1(US)	11/04/209	12/613166	023753/0422
NC27385US	800.0881.U1(US)	1/4/2000	09/477457	010774/0891
NC27385US	800.0881.U2(US)	9/12/2005	11/224718	010774/0891
NC27385US	800.0881.U3(US)	2/26/2009	12/393589	010774/0891
NC33221US	800.0882.U1(US)	11/26/2002	10/304776	013719/0203
NC36695US	800.0883.U1(US)	5/26/2004	10/854532	015080/0994
NC36932US	800.0884.U1(US)	6/28/2004	10/878503	015146/0361
NC40581US	800.0885.U1(US)	8/31/2007	11/885531	020752/0281
NC42803US	800.0886.U1(US)	9/26/2007	11/791201	020810/0762
NC43371US	800.0887.U1(US)	6/21/2006	10/583789	019284/0291
NC45322US	800.0888.U1(US)	8/31/2007	11/885530	020628/0779 021153/0787
NC48411US	800.0889.U1(US)	5/28/2008	11/922372	021010/0691
NC48412US	800.0890.U1(US)	2/29/2008	11/991330	021052/0814
NC48891US	800.0891.U1(US)	7/15/2005	11/182492	016530/0576
NC49025US	800.0892.U1(US)	6/16/2005	11/154246	016521/0396
NC49423US	800.0893.U1(US)	12/27/2005	11/319026	017131/0953
NC49775US	800.0894.U1(US)	12/29/2005	11/321472	017179/0358
NC55083US	800.0896.U1(US)	4/17/2007	11/787474	019308/0364
NC61362US	800.0897.U1(US)	12/20/2007	12/004197	020479/0036
NC64176US	800.0898.U1(US)	7/3/2008	12/217501	21313/0990
NC66708US	800.0899.U1(US)	5/12/2009	12/464414	022873/0442
NC73025US	800.0900.U1(US)	7/1/2010	12/828571	024624/0062
NC16568US	800.0901.U1(US)	8/4/2003	10/634734	014872/0573
NC33132US	800.0902.U1(US)	12/24/2002	10/328858	013876/0230
NC36368US	800.0903.U1(US)	7/6/2004	10/887474	015932/0321
NC36562US	800.0904.U1(US)	9/26/2003	10/673059	015021/0891
NC37226US	800.0905.U1(US)	5/7/2009	12/085750	022651/0956
NC37394US	800.0906.U1(US)	12/4/2006	11/628634	018690/0984
NC43179US	800.0908.U1(US)	5/3/2004	10/838137	15104/0022
NC43689US	800.0909.U1(US)	3/17/2005	11/091364	016707/0469

Nokia Ref.	H&S Ref.	Filing Date	Application No.	Assignment Reel/Frame
NC43786US	800.0910.U1(US)	3/16/2004	10/802391	014888/0309
NC44962US	800.0911.U1(US)	12/22/2005	11/317914	017312/0738
NC46584US	800.0912.U1(US)	2/7/2006	11/350394	017429/0179
NC46695US	800.0913.U1(US)	5/30/2006	11/444156	018251/0311
NC46868US	800.0914.U1(US)	11/3/2006	11/592746	018915/0533
NC46869US	800.0915.U1(US)	11/3/2006	11/592747	018858/0282
NC46968US	800.0916.U1(US)	1/18/2007	11/655763	019327/0770
NC47622US	800.0917.U1(US)	3/14/2006	11/376611	017939/0734
NC47637US	800.0918.U1(US)	5/8/2006	11/431138	018145/0921
NC48088US	800.0919.U1(US)	3/7/2005	11/082633	016396/0558
NC48760US	800.0920.U1(US)	6/15/2005	11/154934	017042/0273
NC48884US	800.0921.U1(US)	11/28/2007	11/998355	016562/0918
NC51149US	800.0922.U1(US)	10/26/2006	11/588187	018613/0587
NC52808US	800.0923.U1(US)	10/25/2007	11/978155	020952/0137
NC52893US	800.0924.U1(US)	8/23/2007	11/895616	020142/0778
NC53831US	800.0925.U1(US)	8/18/2008	12/229065	021813/0805
NC54127US	800.0926.U1(US)	9/18/2007	11/901952	020248/0811
NC54295US	800.0927.U1(US)	9/21/2007	11/903402	020063/0665
NC54327US	800.0928.U1(US)	10/9/2007	11/973882	0205008/0905
NC54512US	800.0929.U1(US)	10/16/2007	11/975138	020332/0257
NC54776US	800.0930.U1(US)	5/24/2007	11/807170	019794/0008
NC55337US	800.0931.U1(US)	3/2/2009	12/227809	22334/0368
NC57291US	800.0932.U1(US)	4/27/2007	11/796620	019509/0679
NC61821US	800.0933.U1(US)	9/8/2008	12/206231	022087/0673
NC63258US	800.0934.U1(US)	8/26/2008	12/198872	021582/0805
NC63261US	800.0935.U1(US)	8/26/2008	12/198839	021582/0582
NC51923US	800.0944.U1(US)	6/14/2010	12/808168	024533/0556
NC73008US	800.0949.U1(US)	8/16/2012	12/867519	24843/0741
NC55214US	800.0950.U1(US)	9/28/2009	12/593361	023630/0802
NC35757US	846A.0035.U1(US)	9/10/2003	10/659777	014821/0686
NC35984US	846A.0036.U1(US)	11/12/2003	10/712232	014992/0126
NC36887US	846A.0037.U1(US)	12/30/2003	10/748088	015308/0309
NC44053US	846A.0038.U1(US)	1/26/2005	11/043730	017658/0616
NC44956US	846A.0039.U1(US)	10/15/2004	10/966747	015523/0298
NC49129US	846A.0040.U1(US)	7/27/2005	11/191152	016481/0357
NC51560US	846A.0041.U1(US)	8/25/2008	12/224348	022167/0443
NC49798US	848A.0006.U1(US)	12/1/2005	11/292786	017543/0313
NC45272US	868A.0104.U1(US)	3/30/2007	11/664483	020476/0990
NC35273US	879A.0152.U1(US)	1/25/2006	11/338996	013741/0536
NC65416US	879A.0153.U1(US)	12/11/2008	12/332846	022106/0654
NC71853US	879A.0154.U1(US)	12/3/2010	12/960138	025841/0009

OPAP IAP4
APR 25 2012
PATENT & TRADEMARK OFFICE

Statement Under 37 C.F.R. 3.73(b)

Owner: **Nokia Corporation**

Application No.: **See Attached**

Nokia Corporation, a corporation
(Name of Assignee) (Type of Assignee)

states that it is:

the assignee of the entire right, title and interest in:

the patent applications **identified on the attached list** by virtue of assignment(s) from the inventor(s) of the patent application(s).

The assignments were recorded in the United States Patent and Trademark Office on the Reel and Frame numbers listed on the attached list.

The undersigned is authorized to act on behalf of the assignee.

Thomas Arria
Title: Senior IPR Specialist

4/19/12
Date

OPAP IAP4
APR 2 5 2012
PATENT & TRADEMARK OFFICE

Nokia Ref.	H&S Ref.	Filing Date	Application No.	Assignment Reel/Frame
NC52687US	800.0266.U1(US)	10/30/2009	12/598368	024739/0870
NC76407US	800.0785.U1(US)	9/27/2011	13/246246	027696/0511
NC46779US	800.0833.U1(US)	2/24/2009	12/438666	027675/0965
NC17810US	800.0870.U1(US)	6/28/2006	11/478555	018643/0540
NC47719US	800.0871.U1(US)	9/8/2006	11/517884	018274/0732
NC52605US	800.0872.U1(US)	4/24/2007	11/789267	019299/0978
NC53552US	800.0873.U1(US)	12/21/2006	11/614498	018963/0980 019229/0394
NC53989US	800.0874.U1(US)	9/13/2007	11/854985	020157/0796
NC57047US	800.0875.U1(US)	3/19/2008	12/051787	021102/0082
NC60014US	800.0876.U1(US)	5/13/2008	12/120226	021488/0823
NC62647US	800.0877.U1(US)	12/18/2007	11/958967	021184/0418 023346/0810
NC70507US	800.0878.U1(US)	11/04/209	12/613166	023753/0422
NC27385US	800.0881.U1(US)	1/4/2000	09/477457	010774/0891
NC27385US	800.0881.U2(US)	9/12/2005	11/224718	010774/0891
NC27385US	800.0881.U3(US)	2/26/2009	12/393589	010774/0891
NC33221US	800.0882.U1(US)	11/26/2002	10/304776	013719/0203
NC36695US	800.0883.U1(US)	5/26/2004	10/854532	015080/0994
NC36932US	800.0884.U1(US)	6/28/2004	10/878503	015146/0361
NC40581US	800.0885.U1(US)	8/31/2007	11/885531	020752/0281
NC42803US	800.0886.U1(US)	9/26/2007	11/791201	020810/0762
NC43371US	800.0887.U1(US)	6/21/2006	10/583789	019284/0291
NC45322US	800.0888.U1(US)	8/31/2007	11/885530	020628/0779 021153/0787
NC48411US	800.0889.U1(US)	5/28/2008	11/922372	021010/0691
NC48412US	800.0890.U1(US)	2/29/2008	11/991330	021052/0814
NC48891US	800.0891.U1(US)	7/15/2005	11/182492	016530/0576
NC49025US	800.0892.U1(US)	6/16/2005	11/154246	016521/0396
NC49423US	800.0893.U1(US)	12/27/2005	11/319026	017131/0953
NC49775US	800.0894.U1(US)	12/29/2005	11/321472	017179/0358
NC55083US	800.0896.U1(US)	4/17/2007	11/787474	019308/0364
NC61362US	800.0897.U1(US)	12/20/2007	12/004197	020479/0036
NC64176US	800.0898.U1(US)	7/3/2008	12/217501	21313/0990
NC66708US	800.0899.U1(US)	5/12/2009	12/464414	022873/0442
NC73025US	800.0900.U1(US)	7/1/2010	12/828571	024624/0062
NC16568US	800.0901.U1(US)	8/4/2003	10/634734	014872/0573
NC33132US	800.0902.U1(US)	12/24/2002	10/328858	013876/0230
NC36368US	800.0903.U1(US)	7/6/2004	10/887474	015932/0321
NC36562US	800.0904.U1(US)	9/26/2003	10/673059	015021/0891
NC37226US	800.0905.U1(US)	5/7/2009	12/085750	022651/0956
NC37394US	800.0906.U1(US)	12/4/2006	11/628634	018690/0984
NC43179US	800.0908.U1(US)	5/3/2004	10/838137	15104/0022
NC43689US	800.0909.U1(US)	3/17/2005	11/091364	016707/0469

Nokia Ref.	H&S Ref.	Filing Date	Application No.	Assignment Reel/Frame
NC43786US	800.0910.U1(US)	3/16/2004	10/802391	014888/0309
NC44962US	800.0911.U1(US)	12/22/2005	11/317914	017312/0738
NC46584US	800.0912.U1(US)	2/7/2006	11/350394	017429/0179
NC46695US	800.0913.U1(US)	5/30/2006	11/444156	018251/0311
NC46868US	800.0914.U1(US)	11/3/2006	11/592746	018915/0533
NC46869US	800.0915.U1(US)	11/3/2006	11/592747	018858/0282
NC46968US	800.0916.U1(US)	1/18/2007	11/655763	019327/0770
NC47622US	800.0917.U1(US)	3/14/2006	11/376611	017939/0734
NC47637US	800.0918.U1(US)	5/8/2006	11/431138	018145/0921
NC48088US	800.0919.U1(US)	3/7/2005	11/082633	016396/0558
NC48760US	800.0920.U1(US)	6/15/2005	11/154934	017042/0273
NC48884US	800.0921.U1(US)	11/28/2007	11/998355	016562/0918
NC51149US	800.0922.U1(US)	10/26/2006	11/588187	018613/0587
NC52808US	800.0923.U1(US)	10/25/2007	11/978155	020952/0137
NC52893US	800.0924.U1(US)	8/23/2007	11/895616	020142/0778
NC53831US	800.0925.U1(US)	8/18/2008	12/229065	021813/0805
NC54127US	800.0926.U1(US)	9/18/2007	11/901952	020248/0811
NC54295US	800.0927.U1(US)	9/21/2007	11/903402	020063/0665
NC54327US	800.0928.U1(US)	10/9/2007	11/973882	0205008/0905
NC54512US	800.0929.U1(US)	10/16/2007	11/975138	020332/0257
NC54776US	800.0930.U1(US)	5/24/2007	11/807170	019794/0008
NC55337US	800.0931.U1(US)	3/2/2009	12/227809	22334/0368
NC57291US	800.0932.U1(US)	4/27/2007	11/796620	019509/0679
NC61821US	800.0933.U1(US)	9/8/2008	12/206231	022087/0673
NC63258US	800.0934.U1(US)	8/26/2008	12/198872	021582/0805
NC63261US	800.0935.U1(US)	8/26/2008	12/198839	021582/0582
NC51923US	800.0944.U1(US)	6/14/2010	12/808168	024533/0556
NC73008US	800.0949.U1(US)	8/16/2012	12/867519	24843/0741
NC55214US	800.0950.U1(US)	9/28/2009	12/593361	023630/0802
NC35757US	846A.0035.U1(US)	9/10/2003	10/659777	014821/0686
NC35984US	846A.0036.U1(US)	11/12/2003	10/712232	014992/0126
NC36887US	846A.0037.U1(US)	12/30/2003	10/748088	015308/0309
NC44053US	846A.0038.U1(US)	1/26/2005	11/043730	017658/0616
NC44956US	846A.0039.U1(US)	10/15/2004	10/966747	015523/0298
NC49129US	846A.0040.U1(US)	7/27/2005	11/191152	016481/0357
NC51560US	846A.0041.U1(US)	8/25/2008	12/224348	022167/0443
NC49798US	848A.0006.U1(US)	12/1/2005	11/292786	017543/0313
NC45272US	868A.0104.U1(US)	3/30/2007	11/664483	020476/0990
NC35273US	879A.0152.U1(US)	1/25/2006	11/338996	013741/0536
NC65416US	879A.0153.U1(US)	12/11/2008	12/332846	022106/0654
NC71853US	879A.0154.U1(US)	12/3/2010	12/960138	025841/0009

OPAP IAP4
APR 25 2012
PATENT & TRADEMARK OFFICE

NOKIA

LIMITED POWER OF ATTORNEY

that we, the undersigned, under the authority granted to each of us to sign jointly on behalf of

NOKIA CORPORATION
Keilalahdentie 4
02150 Espoo
Finland

hereby authorize the following person

Thomas Arria
Nokia Corporation
5 Wayside Road
01803 Burlington
U.S.A.

Nokia Corporation
P.O.Box 226,
FI-00045 NOKIA GROUP, Finland
(Keilalahdentie 4, 02150 ESPOO)

Tel. +358 7180 08000
Fax +358 7180 38226

Business Identity Code
0112038-9, Helsinki

in his capacity as Senior IP Specialist, Patenting, to sign alone in the name of Nokia Corporation all the documents that relate to the filing, prosecution and registration of patent applications, as well as post-grant operations relating to patents granted based on such patent applications. Such documents may include without limitation case-specific Powers of Attorney, Assignment Deeds for assigning the ownership of an invention to Nokia, Novelty Declarations and Inventorship Declarations.

The rights granted under this Power of Attorney may not be transferred further, other than as specified above.

The authority granted by this Power of Attorney shall expire six (6) months from the date of execution.

Notwithstanding the foregoing, the authority granted by this Power of Attorney shall expire when the authorized person's employment with Nokia ends.

This Power of Attorney is duly signed on this day of January 1, 2012.

NOKIA CORPORATION

By:
Name: Harri Honkasalo
Title: Director,
Legal and Intellectual Property

By:
Name: Paul Melin
Title: Vice President,
Legal and Intellectual Property

OPAP LAP4
APR 25 2012
PATENT & TRADEMARK OFFICE

IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(s): HANNUKSELA, Miska

SERIAL NO.: 11/338996

FILED: January 25, 2006

TITLE: Grouping of Image Frames in Video Coding

ATTORNEY DOCKET NO.: 879A.0152.U1 (US)

Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Submission of Revocation of Power Of Attorney
With New Power Of Attorney and
Change of Correspondence Address

Sir:

Attached, please find a copy of the following for the above-identified patent application:

1. Revocation of Power Of Attorney With New Power Of Attorney and Change of Correspondence Address signed by a person authorized to act on behalf of the assignee; and

2. Statement Under 37 C.F.R. 3.73(b).

Please change the power of attorney information and correspondence information for the patent application identified above.

Respectfully submitted,

Mark F. Harrington (Reg. No. 31,686) 4/20/2012
Date

Customer No.: 10948
Harrington & Smith, Attorneys At Law, LLC
4 Research Drive
Shelton, CT 06484-6212
203-925-9400

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
11/338,996	01/25/2006	Miska Hannuksela	879A.0152.U1

CONFIRMATION NO. 5361

10948
Harrington & Smith, Attorneys At Law, LLC
4 Research Drive, Suite 202
Shelton, CT 06484

POA ACCEPTANCE LETTER

OC000000054274531

Date Mailed: 05/15/2012

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 04/25/2012.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/rgtekle/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
11/338,996	01/25/2006	Miska Hannuksela	TPO.004.C1

CONFIRMATION NO. 5361

10888
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

POWER OF ATTORNEY NOTICE

OC000000054274528

Date Mailed: 05/15/2012

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 04/25/2012.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/rgtekle/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

IFW

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

10888 7590 03/19/2012
Hollingsworth & Funk
8500 Normandale Lake Blvd., Suite 320
Minneapolis, MN 55437

OPAP IAP4 MAY 17 2012 PATENT & TRADEMARK OFFICE

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

Elaine F. Mian	(Depositor's name)
Elaine F. Mian	(Signature)
5/14/2012	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,996	01/25/2006	Miska Hannuksela	TPO.004.C1	5361

TITLE OF INVENTION: GROUPING OF IMAGE FRAMES IN VIDEO CODING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1740	$300	$0	$2040	06/19/2012

05/18/2012 MBELETE2 00000002 501924 11338996
01 FC:1501 1740.00 DA
02 FC:1504 300.00 DA

EXAMINER	ART UNIT	CLASS-SUBCLASS
LEE, SIU M	2611	375-240270

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ☒ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 Harrington & Smith
2
3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: Nokia Corporation

(B) RESIDENCE: (CITY and STATE OR COUNTRY): Espoo, Finland

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
- ☒ Issue Fee
- ☒ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number 50-1924 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date May 14, 2012

Typed or printed name Jerry Stanton Registration No. 46,008

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/338,996	06/19/2012	8204134	879A.0152.U1	5361

10948 7590 05/30/2012

Harrington & Smith, Attorneys At Law, LLC
4 Research Drive, Suite 202
Shelton, CT 06484

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 1879 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Miska Hannuksela, Tampere, FINLAND;