8457761

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

February 09, 2024

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: ***13/666,680***
FILING DATE: ***November 01, 2012***
PATENT NUMBER: ***9571833***
ISSUE DATE: ***February 14, 2017***

UNITED STATES PATENT AND TRADEMARK OFFICE · DEPARTMENT OF COMMERCE

Certified by

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

PTO/SB/14 (07-07)
Approved for use through 06/30/2010. 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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NC77198US-NP
	Application Number	
Title of Invention	METHOD FOR CODING AND AN APPARATUS	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Applicant Information:

Applicant 1 Remove

Applicant Authority ◉ Inventor ○ Legal Representative under 35 U.S.C. 117 ○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Mehmet	Oguz	BICI	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Tampere	Country Of Residence[i]	TR

Citizenship under 37 CFR 1.41(b) [i] FI

Mailing Address of Applicant:

Address 1	Tammelan puistokatu 1-3 D 46		
Address 2			
City	Tampere	State/Province	
Postal Code	33500	Country[i]	FI

Applicant 2 Remove

Applicant Authority ◉ Inventor ○ Legal Representative under 35 U.S.C. 117 ○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Jani		LAINEMA	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Tampere	Country Of Residence[i]	FI

Citizenship under 37 CFR 1.41(b) [i] FI

Mailing Address of Applicant:

Address 1	Kisakentankatu 12 B 6		
Address 2			
City	Tampere	State/Province	
Postal Code	33230	Country[i]	FI

Applicant 3 Remove

Applicant Authority ◉ Inventor ○ Legal Representative under 35 U.S.C. 117 ○ Party of Interest under 35 U.S.C. 118

Prefix	Given Name	Middle Name	Family Name	Suffix
	Kemal		UGUR	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Tampere	Country Of Residence[i]	FI

PTO/SB/14 (07-07)
Approved for use through 06/30/2010. 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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NC77198US-NP
	Application Number	
Title of Invention	METHOD FOR CODING AND AN APPARATUS	

Citizenship under 37 CFR 1.41(b) [i]	TR		
Mailing Address of Applicant:			
Address 1	Lapintie 6D 25		
Address 2			
City	Tampere	**State/Province**	
Postal Code	33100	**Country**[i]	FI

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	73658		
Email Address	ipr.inhouse@nokia.com	Add Email	Remove Email

Application Information:

Title of the Invention	METHOD FOR CODING AND AN APPARATUS		
Attorney Docket Number	NC77198US-NP	**Small Entity Status Claimed**	☐
Application Type	Nonprovisional		
Subject Matter	Utility		
Suggested Class (if any)		**Sub Class (if any)**	
Suggested Technology Center (if any)			
Total Number of Drawing Sheets (if any)	13	**Suggested Figure for Publication (if any)**	

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Enter either Customer Number or complete the Representative Name section below. If both sections are completed the Customer Number will be used for the Representative Information during processing.

Please Select One:	◉ Customer Number	○ US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)

PTO/SB/14 (07-07)
Approved for use through 06/30/2010. 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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NC77198US-NP
	Application Number	
Title of Invention	METHOD FOR CODING AND AN APPARATUS	

Customer Number	73658

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78(a)(2) or CFR 1.78(a)(4), and need not otherwise be made part of the specification.

Prior Application Status			Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	non provisional of	61/555703	2011-11-04

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

Foreign Priority Information:

This section allows for the applicant to claim benefit of foreign priority and to identify any prior foreign application for which priority is not claimed. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(a).

Remove

Application Number	Country [i]	Parent Filing Date (YYYY-MM-DD)	Priority Claimed
			◉ Yes ○ No

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Assignee Information:

Providing this information in the application data sheet does not substitute for compliance with any requirement of part 3 of Title 37 of the CFR to have an assignment recorded in the Office.

Assignee 1 Remove

If the Assignee is an Organization check here. ☒

Organization Name	Nokia Corporation		
Mailing Address Information:			
Address 1	Keilalahdentie 4		
Address 2			
City	Espoo	**State/Province**	
Country [i]	FI	Postal Code	02150
Phone Number		Fax Number	
Email Address			

Additional Assignee Data may be generated within this form by selecting the **Add** button. Add

Signature:

A signature of the applicant or representative is required in accordance with 37 CFR 1.33 and 10.18. Please see 37 CFR 1.4(d) for the form of the signature.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000004

PTO/SB/14 (07-07)
Approved for use through 06/30/2010. 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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NC77198US-NP
	Application Number	
Title of Invention	METHOD FOR CODING AND AN APPARATUS	

Signature	/Thomas J. Arria/			Date (YYYY-MM-DD)	2012-11-01
First Name	Thomas J.	Last Name	Arria	Registration Number	60223

This collection of information is required by 37 CFR 1.76. 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 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet 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.**

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000005

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 the Freedom of Information Act requires disclosure of these records.

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 inspections 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.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000006

METHOD FOR CODING AND AN APPARATUS

TECHNICAL FIELD

[0001] There is provided a method for encoding, a method for decoding, an apparatus, computer program products, an encoder and a decoder.

BACKGROUND INFORMATION

[0002] This section is intended to provide a background or context to the invention that is recited in the claims. The description herein may include concepts that could be pursued, but are not necessarily ones that have been previously conceived or pursued. Therefore, unless otherwise indicated herein, what is described in this section is not prior art to the description and claims in this application and is not admitted to be prior art by inclusion in this section

[0003] A video codec may comprise an encoder which transforms input video into a compressed representation suitable for storage and/or transmission and a decoder that can uncompress the compressed video representation back into a viewable form, or either one of them. The encoder may discard some information in the original video sequence in order to represent the video in a more compact form, for example at a lower bit rate.

[0004] Many hybrid video codecs, operating for example according to the International Telecommunication Union's ITU-T H.263 and H.264 coding standards, encode video information in two phases. In the first phase, pixel values in a certain picture area or "block" are predicted. These pixel values can be predicted, for example, by motion compensation mechanisms, which involve finding and indicating an area in one of the previously encoded video frames (or a later coded video frame) that corresponds closely to the block being coded. Additionally, pixel values can be predicted by spatial mechanisms which involve finding and indicating a spatial region relationship, for example by using pixel values around the block to be coded in a specified manner.

[0005] Prediction approaches using image information from a previous (or a later) image can also be called as Inter prediction methods, and prediction approaches using image information within the same image can also be called as Intra prediction methods.

[0006] The second phase is one of coding the error between the predicted block of pixels and the original block of pixels. This may be accomplished by transforming the difference in

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000007

pixel values using a specified transform. This transform may be e.g. a Discrete Cosine Transform (DCT) or a variant thereof. After transforming the difference, the transformed difference may be quantized and entropy encoded.

[0007] By varying the fidelity of the quantization process, the encoder can control the balance between the accuracy of the pixel representation, (in other words, the quality of the picture) and the size of the resulting encoded video representation (in other words, the file size or transmission bit rate).

[0008] The decoder reconstructs the output video by applying a prediction mechanism similar to that used by the encoder in order to form a predicted representation of the pixel blocks (using the motion or spatial information created by the encoder and stored in the compressed representation of the image) and prediction error decoding (the inverse operation of the prediction error coding to recover the quantized prediction error signal in the spatial domain).

[0009] After applying pixel prediction and error decoding processes the decoder combines the prediction and the prediction error signals (the pixel values) to form the output video frame.

[0010] The decoder (and encoder) may also apply additional filtering processes in order to improve the quality of the output video before passing it for display and/or storing as a prediction reference for the forthcoming frames in the video sequence.

[0011] In some video codecs, such as High Efficiency Video Coding Working Draft 4, video pictures may be divided into coding units (CU) covering the area of a picture. A coding unit consists of one or more prediction units (PU) defining the prediction process for the samples within the coding unit and one or more transform units (TU) defining the prediction error coding process for the samples in the coding unit. A coding unit may consist of a square block of samples with a size selectable from a predefined set of possible coding unit sizes. A coding unit with the maximum allowed size can be named as a largest coding unit (LCU) and the video picture may be divided into non-overlapping largest coding units. A largest coding unit can further be split into a combination of smaller coding units, e.g. by recursively splitting the largest coding unit and resultant coding units. Each resulting coding unit may have at least one prediction unit and at least one transform unit associated with it. Each prediction unit and transform unit can further be split into smaller prediction units and transform units in order to increase granularity of the prediction and prediction error coding processes, respectively. Each

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000008

prediction unit may have prediction information associated with it defining what kind of a prediction is to be applied for the pixels within that prediction unit (e.g. motion vector information for inter predicted prediction units and intra prediction directionality information for intra predicted prediction units). Similarly, each transform unit may be associated with information describing the prediction error decoding process for samples within the transform unit (including e.g. discrete cosine transform (DCT) coefficient information). It may be signalled at coding unit level whether prediction error coding is applied or not for each coding unit. In the case there is no prediction error residual associated with the coding unit, it can be considered there are no transform units for the coding unit. The division of the image into coding units, and division of coding units into prediction units and transform units may be signalled in the bitstream allowing the decoder to reproduce the intended structure of these units.

[0012] In some video codecs, motion information is indicated by motion vectors associated with each motion compensated image block. These motion vectors represent the displacement of the image block in the picture to be coded (in the encoder) or decoded (at the decoder) and the prediction source block in one of the previously coded or decoded images (or pictures). In order to represent motion vectors efficiently, motion vectors may be coded differentially with respect to block specific predicted motion vector. In some video codecs, the predicted motion vectors are created in a predefined way, for example by calculating the median of the encoded or decoded motion vectors of the adjacent blocks.

[0013] Another way to create motion vector predictions is to generate a list or a set of candidate predictions from blocks in the current frame and/or co-located or other blocks in temporal reference pictures and signalling the chosen candidate as the motion vector prediction. A spatial motion vector prediction is a prediction obtained only on the basis of information of one or more blocks of the same frame than the current frame whereas temporal motion vector prediction is a prediction obtained on the basis of information of one or more blocks of a frame different from the current frame. It may also be possible to obtain motion vector predictions by combining both spatial and temporal prediction information of one or more encoded blocks. These kinds of motion vector predictions are called as spatio-temporal motion vector predictions.

[0014] In addition to predicting the motion vector values, the reference index in the reference picture list can be predicted. The reference index may be predicted from blocks in the current frame and/or co-located or other blocks in a temporal reference picture. Moreover, some

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000009

high efficiency video codecs employ an additional motion information coding/decoding mechanism, often called merging/merge mode, where all the motion field information, which includes motion vector and corresponding reference picture index for each available reference picture list, may be predicted and used without any modification or correction. Similarly, predicting the motion field information may be carried out using the motion field information of blocks in the current frame and/or co-located or other blocks in temporal reference pictures and the used motion field information is signalled among a list of motion field candidate list filled with motion field information of available blocks in the current frame and/or co-located or other blocks in temporal reference pictures.

[0015] In some video codecs the prediction residual after motion compensation is first transformed with a transform kernel (like DCT) and then coded. The reason for this is that often there still exists some correlation among the residual and transform can in many cases help reduce this correlation and provide more efficient coding.

[0016] Some video encoders utilize Lagrangian cost functions to find optimal coding modes, e.g. the desired Macroblock mode and associated motion vectors. This kind of cost function uses a weighting factor λ to tie together the (exact or estimated) image distortion due to lossy coding methods and the (exact or estimated) amount of information that is required to represent the pixel values in an image area:

$$C = D + \lambda R \quad (1)$$

where C is the Lagrangian cost to be minimized, D is the image distortion (e.g. Mean Squared Error) with the mode and motion vectors considered, and R the number of bits needed to represent the required data to reconstruct the image block in the decoder (including the amount of data to represent the candidate motion vectors).

[0017] Some video codecs such as hybrid video codecs may generate a list of motion vector predictions (MVP) consisting of motion vectors of spatial adjacent blocks (spatial MVP) and/or motion vectors of blocks in a previously decoded frame (temporal MVP). One of the candidate motion vectors in the list is signalled to be used as the motion vector prediction of the current block. After the list is generated, some of the motion vector prediction candidates may have the same motion information. In this case, the identical motion vector prediction candidates

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000010

may be removed to reduce redundancy. During the decoding, if the temporal motion vector prediction information is unavailable due to e.g. loss of reference frame, the decoder may not know if the temporal motion vector prediction candidate in the list is to be removed. This may lead to uncertainty for mapping the decoded candidate index to the candidates whose removal decision is based on comparing motion information with the temporal motion vector prediction. As a result, false assignment of motion vector prediction candidates may occur which may lead to degradation in the picture quality and drift of false motion information throughout the decoding process.

SUMMARY

[0018] The present invention introduces a method for generating a motion vector prediction list for an image block. In some embodiments video codecs employ in a motion prediction candidate list construction a way to reduce the complexity of the implementation. This can be achieved by performing a limited number of motion information comparisons between candidate pairs to remove the redundant candidates rather than comparing every available candidate pair. The decision of whether comparing two candidates may depend on the order of the candidates to be considered for the list and/or coding/prediction mode and/or location of the blocks associated with the candidates. In some embodiments a video codec employs a merge process for motion information coding and creates a list of motion prediction candidates from which one of the candidates is to be signalled as the motion information for the current coding or prediction unit. The motion prediction candidates may consist of several spatial motion predictions and a temporal motion prediction. The spatial candidates are obtained from the motion information of e.g. spatial neighbour blocks.

[0019] According to a first aspect of the present invention there is provided a method comprising:

receiving a block of pixels including a prediction unit; determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000011

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[0020] According to a second aspect of the present invention there is provided a method comprising:

receiving an encoded block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of another spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[0021] According to a third aspect of the present invention there is provided an apparatus comprising a processor and a memory including computer program code, the memory and the computer program code configured to, with the processor, cause the apparatus to:

receive a block of pixels

including a prediction unit;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000012

determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[0022] According to a fourth aspect of the present invention there is provided an apparatus comprising a processor and a memory including computer program code, the memory and the computer program code configured to, with the processor, cause the apparatus to:

receive an encoded block of pixels

including a prediction unit;

determining a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of another spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000013

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[0023] According to a fifth aspect of the present invention there is provided a storage medium having stored thereon a computer executable program code for use by an encoder, said program code comprises instructions for:

receiving a block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

select a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determine a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

compare motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

exclude the first spatial motion vector prediction candidate from the merge list, if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other

[0024] According to a sixth aspect of the present invention there is provided a storage medium having stored thereon a computer executable program code for use by a decoder, said program code comprises instructions for:

receiving an encoded block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000014

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[0025] According to a seventh aspect of the present invention there is provided an apparatus comprising:

means for receiving a block of pixels including a prediction unit;

means for determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[0026] According to an eighth aspect of the present invention there is provided an apparatus comprising:

means for receiving an encoded block of pixels including a prediction unit;

means for determining a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000015

means for selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

means for determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

means for comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

means for excluding the first spatial motion vector prediction candidate from the merge list, if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other.

DESCRIPTION OF THE DRAWINGS

[0027] For better understanding of the present invention, reference will now be made by way of example to the accompanying drawings in which:

[0028] Figure 1 shows schematically an electronic device employing some embodiments of the invention;

[0029] Figure 2 shows schematically a user equipment suitable for employing some embodiments of the invention;

[0030] Figure 3 further shows schematically electronic devices employing embodiments of the invention connected using wireless and wired network connections;

[0031] Figure 4a shows schematically an embodiment of the invention as incorporated within an encoder;

[0032] Figure 4b shows schematically an embodiment of a prediction reference list generation and modification according to some embodiments of the invention;

[0033] Figures 5a and 5b show a flow diagram showing the operation of an embodiment of the invention with respect to the encoder as shown in figure 4a;

[0034] Figure 6a illustrates an example of spatial and temporal prediction of a prediction unit;

[0035] Figure 6b illustrates another example of spatial and temporal prediction of a prediction unit;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000016

[0036] Figure 7 shows schematically an embodiment of the invention as incorporated within a decoder;

[0037] Figures 8a and 8b show a flow diagram of showing the operation of an embodiment of the invention with respect to the decoder shown in figure 7;

[0038] Figure 9 illustrates an example of a coding unit and some neighbour blocks of the coding unit;

[0039] Figure 10a illustrates an example of a horizontal division of the coding unit;

[0040] Figure 10b illustrates an example of a vertical division of the coding unit;

[0041] Figure 11a illustrates locations of five spatial neighbours A0, A1, B0, B1, B2 for a prediction unit generated as the second prediction unit of a horizontally divided coding unit;

[0042] Figure 11b illustrates locations of five spatial neighbours for a prediction unit generated as the second prediction unit of a vertically divided coding unit; and

[0043] Figure 12 illustrates an example of blocks between some spatial neighbours of a coding unit.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000017

DETAILED DESCRIPTION OF SOME EXAMPLE EMBODIMENTS

[0044] The following describes in further detail suitable apparatus and possible mechanisms for the provision of improving the prediction accuracy and hence possibly reducing information to be transmitted in video coding systems. In this regard reference is first made to Figure 1 which shows a schematic block diagram of an exemplary apparatus or electronic device 50, which may incorporate a codec according to an embodiment of the invention.

[0045] The electronic device 50 may for example be a mobile terminal or user equipment of a wireless communication system. However, it would be appreciated that embodiments of the invention may be implemented within any electronic device or apparatus which may require encoding and decoding or encoding or decoding video images.

[0046] The apparatus 50 may comprise a housing 30 for incorporating and protecting the device. The apparatus 50 further may comprise a display 32 in the form of a liquid crystal display. In other embodiments of the invention the display may be any suitable display technology suitable to display an image or video. The apparatus 50 may further comprise a keypad 34. In other embodiments of the invention any suitable data or user interface mechanism may be employed. For example the user interface may be implemented as a virtual keyboard or data entry system as part of a touch-sensitive display. The apparatus may comprise a microphone 36 or any suitable audio input which may be a digital or analogue signal input. The apparatus 50 may further comprise an audio output device which in embodiments of the invention may be any one of: an earpiece 38, speaker, or an analogue audio or digital audio output connection. The apparatus 50 may also comprise a battery 40 (or in other embodiments of the invention the device may be powered by any suitable mobile energy device such as solar cell, fuel cell or clockwork generator). The apparatus may further comprise an infrared port 42 for short range line of sight communication to other devices. In other embodiments the apparatus 50 may further comprise any suitable short range communication solution such as for example a Bluetooth wireless connection or a USB/firewire wired connection.

[0047] The apparatus 50 may comprise a controller 56 or processor for controlling the apparatus 50. The controller 56 may be connected to memory 58 which in embodiments of the invention may store both data in the form of image and audio data and/or may also store instructions for implementation on the controller 56. The controller 56 may further be connected

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000018

to codec circuitry 54 suitable for carrying out coding and decoding of audio and/or video data or assisting in coding and decoding carried out by the controller 56.

[0048] The apparatus 50 may further comprise a card reader 48 and a smart card 46, for example a UICC and UICC reader for providing user information and being suitable for providing authentication information for authentication and authorization of the user at a network.

[0049] The apparatus 50 may comprise radio interface circuitry 52 connected to the controller and suitable for generating wireless communication signals for example for communication with a cellular communications network, a wireless communications system or a wireless local area network. The apparatus 50 may further comprise an antenna 44 connected to the radio interface circuitry 52 for transmitting radio frequency signals generated at the radio interface circuitry 52 to other apparatus(es) and for receiving radio frequency signals from other apparatus(es).

[0050] In some embodiments of the invention, the apparatus 50 comprises a camera capable of recording or detecting individual frames which are then passed to the codec 54 or controller for processing. In some embodiments of the invention, the apparatus may receive the video image data for processing from another device prior to transmission and/or storage. In some embodiments of the invention, the apparatus 50 may receive either wirelessly or by a wired connection the image for coding/decoding.

[0051] With respect to Figure 3, an example of a system within which embodiments of the present invention can be utilized is shown. The system 10 comprises multiple communication devices which can communicate through one or more networks. The system 10 may comprise any combination of wired or wireless networks including, but not limited to a wireless cellular telephone network (such as a GSM, UMTS, CDMA network etc), a wireless local area network (WLAN) such as defined by any of the IEEE 802.x standards, a Bluetooth personal area network, an Ethernet local area network, a token ring local area network, a wide area network, and the Internet.

[0052] The system 10 may include both wired and wireless communication devices or apparatus 50 suitable for implementing embodiments of the invention.

[0053] For example, the system shown in Figure 3 shows a mobile telephone network 11 and a representation of the internet 28. Connectivity to the internet 28 may include, but is not

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000019

limited to, long range wireless connections, short range wireless connections, and various wired connections including, but not limited to, telephone lines, cable lines, power lines, and similar communication pathways.

[0054] The example communication devices shown in the system 10 may include, but are not limited to, an electronic device or apparatus 50, a combination of a personal digital assistant (PDA) and a mobile telephone 14, a PDA 16, an integrated messaging device (IMD) 18, a desktop computer 20, a notebook computer 22. The apparatus 50 may be stationary or mobile when carried by an individual who is moving. The apparatus 50 may also be located in a mode of transport including, but not limited to, a car, a truck, a taxi, a bus, a train, a boat, an airplane, a bicycle, a motorcycle or any similar suitable mode of transport.

[0055] Some or further apparatuses may send and receive calls and messages and communicate with service providers through a wireless connection 25 to a base station 24. The base station 24 may be connected to a network server 26 that allows communication between the mobile telephone network 11 and the internet 28. The system may include additional communication devices and communication devices of various types.

[0056] The communication devices may communicate using various transmission technologies including, but not limited to, code division multiple access (CDMA), global systems for mobile communications (GSM), universal mobile telecommunications system (UMTS), time divisional multiple access (TDMA), frequency division multiple access (FDMA), transmission control protocol-internet protocol (TCP-IP), short messaging service (SMS), multimedia messaging service (MMS), email, instant messaging service (IMS), Bluetooth, IEEE 802.11 and any similar wireless communication technology. A communications device involved in implementing various embodiments of the present invention may communicate using various media including, but not limited to, radio, infrared, laser, cable connections, and any suitable connection.

[0057] With respect to Figure 4a, a block diagram of a video encoder suitable for carrying out embodiments of the invention is shown. Furthermore, with respect to Figures 5a and 5b, the operation of the encoder exemplifying embodiments of the invention specifically with respect to construction of the list of candidate predictions is shown as a flow diagram.

[0058] Figure 4a shows the encoder as comprising a pixel predictor 302, prediction error encoder 303 and prediction error decoder 304. Figure 4a also shows an embodiment of the pixel

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000020

predictor 302 as comprising an inter-predictor 306, an intra-predictor 308, a mode selector 310, a filter 316, and a reference frame memory 318. In this embodiment the mode selector 310 comprises a block processor 381 and a cost evaluator 382. The encoder may further comprise an entropy encoder 330 for entropy encoding the bit stream.

[0059] Figure 4b depicts an embodiment of the inter predictor 306. The inter predictor 306 comprises a reference frame selector 360 for selecting reference frame or frames, a motion vector definer 361, a prediction list modifier 363 and a motion vector selector 364. These elements or some of them may be part of a prediction processor 362 or they may be implemented by using other means.

[0060] The pixel predictor 302 receives the image 300 to be encoded at both the inter-predictor 306 (which determines the difference between the image and a motion compensated reference frame 318) and the intra-predictor 308 (which determines a prediction for an image block based only on the already processed parts of the current frame or picture). The output of both the inter-predictor and the intra-predictor may be passed to the mode selector 310. The intra-predictor 308 may have more than one intra-prediction modes. Hence, each mode may perform the intra-prediction and provide the predicted signal to the mode selector 310. The mode selector 310 also receives a copy of the image 300.

[0061] The mode selector 310 determines which encoding mode to use to encode the current block. If the mode selector 310 decides to use an inter-prediction mode it will pass the output of the inter-predictor 306 to the output of the mode selector 310. If the mode selector 310 decides to use an intra-prediction mode it will pass the output of one of the intra-predictor modes to the output of the mode selector 310.

[0062] The output of the mode selector is passed to a first summing device 321. The first summing device may subtract the pixel predictor 302 output from the image 300 to produce a first prediction error signal 320 which is input to the prediction error encoder 303.

[0063] The pixel predictor 302 further receives from a preliminary reconstructor 339 the combination of the prediction representation of the image block 312 and the output 338 of the prediction error decoder 304. The preliminary reconstructed image 314 may be passed to the intra-predictor 308 and to a filter 316. The filter 316 receiving the preliminary representation may filter the preliminary representation and output a final reconstructed image 340 which may be saved in a reference frame memory 318. The reference frame memory 318 may be connected

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000021

to the inter-predictor 306 to be used as the reference image against which the future image 300 is compared in inter-prediction operations.

[0064] The operation of the pixel predictor 302 may be configured to carry out any known pixel prediction algorithm known in the art.

[0065] The pixel predictor 302 may also comprise a filter 385 to filter the predicted values before outputting them from the pixel predictor 302.

[0066] The operation of the prediction error encoder 302 and prediction error decoder 304 will be described hereafter in further detail. In the following examples the encoder generates images in terms of 16x16 pixel macroblocks which go to form the full image or picture. Thus, for the following examples the pixel predictor 302 outputs a series of predicted macroblocks of size 16x16 pixels and the first summing device 321 outputs a series of 16x16 pixel residual data macroblocks which may represent the difference between a first macro-block in the image 300 against a predicted macro-block (output of pixel predictor 302). It would be appreciated that other size macro blocks may be used.

[0067] The prediction error encoder 303 comprises a transform block 342 and a quantizer 344. The transform block 342 transforms the first prediction error signal 320 to a transform domain. The transform is, for example, the DCT transform. The quantizer 344 quantizes the transform domain signal, e.g. the DCT coefficients, to form quantized coefficients.

[0068] The prediction error decoder 304 receives the output from the prediction error encoder 303 and performs the opposite processes of the prediction error encoder 303 to produce a decoded prediction error signal 338 which when combined with the prediction representation of the image block 312 at the second summing device 339 produces the preliminary reconstructed image 314. The prediction error decoder may be considered to comprise a dequantizer 346, which dequantizes the quantized coefficient values, e.g. DCT coefficients, to reconstruct the transform signal and an inverse transformation block 348, which performs the inverse transformation to the reconstructed transform signal wherein the output of the inverse transformation block 348 contains reconstructed block(s). The prediction error decoder may also comprise a macroblock filter (not shown) which may filter the reconstructed macroblock according to further decoded information and filter parameters.

[0069] In the following the operation of an example embodiment of the inter predictor 306 will be described in more detail. The inter predictor 306 receives the current block for inter

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000022

prediction. It is assumed that for the current block there already exists one or more neighbouring blocks which have been encoded and motion vectors have been defined for them. For example, the block on the left side and/or the block above the current block may be such blocks. Spatial motion vector predictions for the current block can be formed e.g. by using the motion vectors of the encoded neighbouring blocks and/or of non-neighbour blocks in the same slice or frame, using linear or non-linear functions of spatial motion vector predictions, using a combination of various spatial motion vector predictors with linear or non-linear operations, or by any other appropriate means that do not make use of temporal reference information. It may also be possible to obtain motion vector predictors by combining both spatial and temporal prediction information of one or more encoded blocks. These kinds of motion vector predictors may also be called as spatio-temporal motion vector predictors.

[0070] Reference frames used in encoding the neighbouring blocks have been stored to the reference frame memory 404. The reference frames may be short term references or long term references and each reference frame may have a unique index indicative of the location of the reference frame in the reference frame memory. When a reference frame is no longer used as a reference frame it may be removed from the reference frame memory or marked as a non-reference frame wherein the storage location of that reference frame may be occupied for a new reference frame. In addition to the reference frames of the neighbouring blocks the reference frame selector 360 may also select one or more other frames as potential reference frames and store them to the reference frame memory.

[0071] Motion vector information of encoded blocks is also stored into the memory so that the inter predictor 306 is able to retrieve the motion vector information when processing motion vector candidates for the current block.

[0072] In some embodiments the motion vectors are stored into one or more lists. For example, motion vectors of uni-directionally predicted frames (e.g. P-frames) may be stored to a list called as list 0. For bi-directionally predicted frames (e.g. B-frames) there may be two lists (list 0 and list 1) and for multi-predicted frames there may be more than two lists. Reference frame indices possibly associated with the motion vectors may also be stored in one or more lists.

[0073] In some embodiments there may be two or more motion vector prediction procedures and each procedure may have its own candidate set creation process. In one

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000023

procedure, only the motion vector values are used. In another procedure, which may be called as a Merge Mode, each candidate element may comprise 1) The information whether 'block was uni-predicted using only list0' or 'block was uni-predicted using only list1' or 'block was bi-predicted using list0 and list1' 2) motion vector value for list0 3) Reference picture index in list0 4) motion vector value for list1 5) Reference picture index list1. Therefore, whenever two prediction candidates are to be compared, not only the motion vector values are compared, but also the five values mentioned above may be compared to determine whether they correspond with each other or not. On the other hand, if any of the comparisons indicate that the prediction candidates do not have equal motion information, no further comparisons need be performed.

[0074] The motion vector definer 361 defines candidate motion vectors for the current frame by using one or more of the motion vectors of one or more neighbour blocks and/or other blocks of the current block in the same frame and/or co-located blocks and/or other blocks of the current block in one or more other frames. These candidate motion vectors can be called as a set of candidate predictors or a predictor set. Each candidate predictor thus represents the motion vector of one or more already encoded block. In some embodiments the motion vector of the candidate predictor is set equal to the motion vector of a neighbour block for the same list if the current block and the neighbour block refer to the same reference frames for that list. Also for temporal prediction there may be one or more previously encoded frames wherein motion vectors of a co-located block or other blocks in a previously encoded frame can be selected as candidate predictors for the current block. The temporal motion vector predictor candidate can be generated by any means that make use of the frames other than the current frame.

[0075] The candidate motion vectors can also be obtained by using more than one motion vector of one or more other blocks such as neighbour blocks of the current block and/or co-located blocks in one or more other frames. As an example, any combination of the motion vector of the block to the left of the current block, the motion vector of the block above the current block, and the motion vector of the block at the up-right corner of the current block may be used (i.e. the block to the right of the block above the current block). The combination may be a median of the motion vectors or calculated by using other formulas. For example, one or more of the motion vectors to be used in the combination may be scaled by a scaling factor, an offset may be added, and/or a constant motion vector may be added. In some embodiments the combined motion vector is based on both temporal and spatial motion vectors, e.g. the motion

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000024

vector of one or more of the neighbour block or other block of the current block and the motion vector of a co-located block or other block in another frame.

[0076] If a neighbour block does not have any motion vector information a default motion vector such as a zero motion vector may be used instead.

[0077] Figure 9 illustrates an example of a coding unit 900 and some neighbour blocks 901—905 of the coding unit. As can be seen from Figure 9, if the coding unit 900 represents the current block, the neighbouring blocks 901—905 labelled A0, A1, B0, B1 and B2 could be such neighbour blocks which may be used when obtaining the candidate motion vectors.

[0078] Creating additional or extra motion vector predictions based on previously added predictors may be needed when the current number of candidates is limited or insufficient. This kind of creating additional candidates can be performed by combining previous two predictions and/or processing one previous candidate by scaling or adding offset and/or adding a zero motion vector with various reference indices. Hence, the motion vector definer 361 may examine how many motion vector candidates can be defined and how many potential candidate motion vectors exist for the current block. If the number of potential motion vector candidates is smaller than a threshold, the motion vector definer 361 may create additional motion vector predictions.

[0079] In some embodiments the combined motion vector can be based on motion vectors in different lists. For example, one motion vector may be defined by combining one motion vector from the list 0 and one motion vector from the list 1 e.g. when the neighbouring or co-located block is a bi-directionally predicted block and there exists one motion vector in the list 0 and one motion vector in the list 1 for the bi-directionally predicted block.

[0080] To distinguish the current block from the encoded/decoded blocks the motion vectors of which are used as candidate motion vectors, those encoded/decoded blocks are also called as reference blocks in this application.

[0081] In some embodiments not only the motion vector information of the reference block(s) is obtained (e.g. by copying) but also a reference index of the reference block in the reference picture list may be copied to the candidate list. The information whether the block was uni-predicted using only list0 or the block was uni-predicted using only list1 or the block was bi-predicted using list0 and list1 may also be copied. The candidate list may also be called as a candidate set or a set of motion vector prediction candidates.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000025

[0082] Figure 6a illustrates an example of spatial and temporal prediction of a prediction unit. There is depicted the current block 601 in the frame 600 and a neighbour block 602 which already has been encoded. The motion vector definer 361 has defined a motion vector 603 for the neighbour block 602 which points to a block 604 in the previous frame 605. This motion vector can be used as a potential spatial motion vector prediction 610 for the current block. Figure 6a depicts that a co-located block 606 in the previous frame 605, i.e. the block at the same location than the current block but in the previous frame, has a motion vector 607 pointing to a block 609 in another frame 608. This motion vector 607 can be used as a potential temporal motion vector prediction-611 for the current frame.

[0083] Figure 6b illustrates another example of spatial and temporal prediction of a prediction unit. In this example the block 606 of the previous frame 605 uses bi-directional prediction based on the block 609 of the frame preceding the frame 605 and on the block 612 succeeding the current frame 600. The temporal motion vector prediction for the current block 601 may be formed by using both the motion vectors 607, 614 or either of them.

[0084] The operation of the prediction list modifier 363 will now be described in more detail with reference to the flow diagram of Figures 5a and 5b. The prediction list modifier 363 initializes a motion vector prediction list to default values in block 500 of Figure 5a. The prediction list modifier 363 may also initialize a list index to an initial value such as zero. Then, in block 501 the prediction list modifier checks whether there are any motion vector candidates to process. If there is at least one motion vector candidate in the predictor set for processing, the prediction list modifier 363 generates the next motion vector candidate which may be a temporal motion vector or a spatial motion vector. The comparison can be an identicality/equivalence check or comparing the (absolute) difference against a threshold or any other similarity metric.

[0085] In the following, a merge process for motion information coding according to an example embodiment will be described in more detail. The encoder creates a list of motion prediction candidates from which one of the candidates is to be signalled as the motion information for the current coding unit or prediction unit. The motion prediction candidates may consist of several spatial motion predictions and a temporal motion prediction. The spatial candidates can be obtained from the motion information of e.g. the spatial neighbour blocks A0, A1, B0, B1, B2, whose motion information is used as spatial candidate motion predictions. The temporal motion prediction candidate may be obtained by processing the motion of a block in a

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000026

frame other than the current frame. In this example embodiment, the encoder operations to construct the merge list for the spatial candidates may include the following. The operations may be carried out by the prediction list modifier 363, for example.

[0086] A maximum number of spatial motion prediction candidates to be included in the merge list may be defined. This maximum number may have been stored, for example, to the memory 58 of the apparatus 50, or to another appropriate place. It is also possible to determine the maximum number by using other means, or it may be determined in the software of the encoder of the apparatus 50.

[0087] In some embodiments the maximum number of spatial motion prediction candidates to be included in the merge list is four but in some embodiments the maximum number may be less than four or greater than four.

[0088] In this example the spatial motion prediction candidates are the spatial neighbour blocks A0, A1, B0, B1, B2. The spatial motion vector prediction candidate A1 is located on the left side of the prediction unit when the encoding/decoding order is from left to right and from top to bottom of the frame, slice or another entity to be encoded/decoded. Respectively, the spatial motion vector prediction candidate B1 is located above the prediction unit. third; the spatial motion vector prediction candidate B0 is on the right side of the spatial motion vector prediction candidate B1; the spatial motion vector prediction candidate A0 is below the spatial motion vector prediction candidate A1; and the spatial motion vector prediction candidate B2 is located on the same column than spatial motion vector prediction candidate A1 and on the same row than the spatial motion vector prediction candidate B1. In other words, the spatial motion vector prediction candidate B2 is cornerwise neighbouring the prediction unit as can be seen e.g. from Figure 9.

[0089] These spatial motion prediction candidates can be processed in a predetermined order, for example, A1, B1, B0, A0 and B2. The first spatial motion prediction candidate to be selected for further examination is thus A1. Before further examination is performed for the selected spatial motion prediction candidate, it may be determined whether the merge list already contains a maximum number of spatial motion prediction candidates. Hence, the prediction list modifier 363 compares 502 the number of spatial motion prediction candidates in the merge list with the maximum number, and if the number of spatial motion prediction candidates in the merge list is not less than the maximum number, the selected spatial motion prediction candidate

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000027

is not included in the merge list and the process of constructing the merge list can be stopped 526. On the other hand, if the number of spatial motion prediction candidates in the merge list is less than the maximum number, a further analyses of the selected spatial motion prediction candidate is performed (blocks 504-522).

[0090] For all the spatial motion prediction candidates for which the further analyses is to be performed, some or all of the following conditions below may be tested for determining whether to include the spatial motion prediction candidate in the merge list.

[0091] The prediction list modifier 363 examines 504 if the prediction unit or block covering the spatial motion prediction candidate block is not available for motion prediction. If so, the candidate is not included in the merge list. The reason that the block is not available may be that the block is either coded in intra mode or resides in a different slice or outside of the picture area.

[0092] In addition to the common conditions above, for each spatial motion prediction candidate, if any of the following conditions holds, then the candidate is not included in the merge list, otherwise, it is included.

[0093] The prediction list modifier 363 determines 506 which spatial motion prediction candidate of the set of spatial motion prediction candidates is in question. If the spatial motion prediction candidate is the block A1, one or more of the following conditions may be examined 508, 510 to determine whether to include this spatial motion prediction candidate in the merge list or not. If the current coding unit 100 is vertically split into two rectangle prediction units 103, 104 as depicted in Figure 10b and the current prediction unit is the second prediction unit 104 in the coding/decoding order (508), this spatial motion prediction candidate is not included in the merge list. If the current coding unit 100 is not vertically split into two rectangle prediction units but it is horizontally split into two rectangle prediction units 101, 102 as depicted in Figure 10a and the current prediction unit is the second prediction unit in the coding/decoding order and the block A1 has the same motion information as the block B1 (510), this spatial motion prediction candidate (block A1) is not included in the merge list. In the example of Figure 10a the second prediction unit is the lower prediction unit 102 of the coding unit 100 and in the example of Figure 10b the second prediction unit is the rightmost prediction unit 104 of the coding unit 100. If none of the conditions above is fulfilled the block A1 is included in the merge list as a spatial motion prediction candidate (524).

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000028

[0094] If the spatial motion prediction candidate is the block B1, one or more of the following conditions may be examined 512, 514 to determine whether to include this spatial motion prediction candidate in the merge list or not. If the current coding unit 100 is horizontally split into two rectangle prediction units 101, 102 as depicted in Figure 10a and the current prediction unit is the second prediction unit 104 in the coding/decoding order (512), this spatial motion prediction candidate is not included in the merge list. If the current coding unit 100 is not horizontally split into two rectangle prediction units and if the block B1 has the same motion information than the block A1 (514), this spatial motion prediction candidate (block B1) is not included in the merge list. If none of the conditions above is fulfilled the block B1 is included in the merge list as a spatial motion prediction candidate (524).

[0095] If the spatial motion prediction candidate is the block B0, this spatial motion prediction candidate is not included in the merge list if the block B0 has the same motion information than the block B1 (516). Otherwise, if the number of spatial motion prediction candidates in the merge list is less than the maximum number of spatial motion prediction candidates, this spatial motion prediction candidate (block B0) is included in the merge list (524).

[0096] If the spatial motion prediction candidate is the block A0, this spatial motion prediction candidate is not included in the merge list if the block A0 has the same motion information than the block A1 (518). Otherwise, if the number of spatial motion prediction candidates in the merge list is less than the maximum number of spatial motion prediction candidates, this spatial motion prediction candidate (block A0) is included in the merge list (524).

[0097] If the spatial motion prediction candidate is the block B2, this spatial motion prediction candidate is not included in the merge list if the maximum number of spatial motion prediction candidates is four and the other blocks A0, A1, B0, and B1 are all decided to be included in the merge list (520). Otherwise, if the number of spatial motion prediction candidates in the merge list is less than the maximum number of spatial motion prediction candidates, the block B2 is not included in the merge list if the block B2 has the same motion information than the block B1 or the block A1 (522).

[0098] Then, after processing the blocks A1, B1, B0, A0 and B2 and including a subset of them in the merge list based on the above described conditions, no more redundancy check

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000029

between these candidates are performed and remaining temporal motion prediction candidate and/or other possible additional candidates may be processed.

[0099] Comparing two blocks whether they have the same motion may be performed by comparing all the elements of the motion information, namely 1) The information whether 'the prediction unit is uni-predicted using only reference picture list0' or 'the prediction unit is uni-predicted using only reference picture list1' or 'the prediction unit is bi-predicted using both reference picture list0 and list1' 2) Motion vector value corresponding to the reference picture list0 3) Reference picture index in the reference picture list0 4) Motion vector value corresponding to the reference picture list1 5) Reference picture index in the reference picture list1.

[00100] In some embodiments similar restrictions for comparing candidate pairs can be applied if the current coding unit is coded/decoded by splitting into four or any number of prediction units.

[00101] The maximum number of merge list candidates can be any non-zero value. In the example above the merger list candidates were the spatial neighbour blocks A0, A1, B0, B1, B2 and the temporal motion prediction candidate, but there may be more than one temporal motion prediction candidate and also other spatial motion prediction candidates than the spatial neighbour blocks. In some embodiments there may also be other spatial neighbour blocks than the blocks A0, A1, B0, B1, B2.

[00102] It is also possible that the maximum number of spatial motion prediction candidates included in the list can be different than four.

[00103] In some embodiments the maximum number of merge list candidates and maximum number of spatial motion prediction candidates included in the list can depend on whether a temporal motion vector candidate is included in the list or not.

[00104] A different number of spatial motion prediction candidates located at various locations in the current frame can be processed. The locations can be the same as or different than A1, B1, B0, A0 and B2.

[00105] The decision of including which spatial motion prediction candidates in the list can be realized in two steps. In the first step, some of the candidates are eliminated by checking whether the candidate block is available and/or the candidate block's prediction mode is intra and/or whether the current block is a second prediction unit of a coding unit coded with two

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000030

prediction units and the candidate has the same motion with the first prediction unit. In the second step, remaining candidates are examined and some or all of them are included in the merge list. The examination in the second step does not include comparing motion information of each possible candidate pair but includes a subset of the possible comparison combinations.

[00106] The decisions for the candidates can be taken in any order of A1, B1, B0, A0 and B2 or independently in parallel.

[00107] For each candidate and/or a subset of the candidates, the following conditions may also be checked: Whether the candidate block has the same motion as the first prediction unit of the current coding unit when the current coding unit is split into two rectangle prediction units and the current prediction unit is the second prediction unit in the coding/decoding order.

[00108] Additional conditions related to various properties of current and/or previous slices and/or current and/or neighbour blocks can be utilized for determining whether to include a candidate in the list.

[00109] Motion comparison can be realized by comparing a subset of the whole motion information. For example, only the motion vector values for some or all reference picture lists and/or reference indices for some or all reference picture lists and/or an identifier value assigned to each block to represent its motion information can be compared. The comparison can be an identicality or an equivalence check or comparing the (absolute) difference against a threshold or any other similarity metric.

[00110] Conditions for deciding whether a candidate is to be included in the list can include motion information comparison with any subset of the candidates as long as not all possible candidate pairs are compared eventually.

[00111] Deciding whether a temporal motion vector candidate is to be included in the list can be based on comparing its motion information with motion information of a subset of the spatial motion vector prediction candidates.

[00112] When comparing motion information of two blocks, motion information of additional blocks can be considered too. For example, when comparing the block B2 and the block A1, all the blocks between the block B2 and the block A1 (illustrated in Figure 12) are checked whether they have the same motion; and when comparing the block B2 and the block B1, all the blocks between the block B2 and the block B1 (illustrated in Figure 12) are checked whether they have the same motion. This embodiment can be implemented so that the right-most

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000031

block of each prediction unit or all blocks of each prediction unit may store the information of how many consecutive blocks to the above have the same motion information. Also the bottom-most block of each prediction unit or all blocks of each prediction unit may store the information of how many consecutive blocks to the left have the same motion information. Using this information the condition for not including B0 in the list can be realized by checking if the number of consecutive blocks with the same motion to the left of B0 is greater than 0. The condition for not including A0 in the list can be realized by checking if the number of consecutive blocks with same motion to the above of A0 is greater than 0. The conditions for not including B2 can be modified as follows:

[00113] It is not examined whether the block B2 has same motion as the block B1 or whether the block B2 has same motion as the block A1, but how many consecutive blocks exists to the left of the block B1 with the same motion than the block B1 and/or how many consecutive blocks exist above the block A1 with the same motion. If the number of consecutive blocks with the same motion to the left of the block B1 is greater than the number of blocks between B2 and B1, or if the number of consecutive blocks with the same motion above the block A1 is greater than the number of blocks between the block B2 and the block A1, the block B2 is not included in the merge list.

[00114] If the above implementation is used, the value of how many consecutive blocks to the left/above have the same motion information can be determined by direct comparison of motion information or checking the prediction mode and/or the merge index if the block employs a merge process.

[00115] When coding/decoding the selected merge index, the information whether the merge process is employed for coding/decoding a Skip mode coding unit or an Inter Merge mode prediction unit can be taken into account. For example, if a context adaptive binary arithmetic coder (CABAC) is used for entropy coding/decoding, different contexts can be used for the bins depending on the coding mode (Skip mode or inter merge mode) of the current block. Furthermore, assigning two contexts depending on whether the merge process is employed in a Skip mode coding unit or an inter Merge mode prediction unit can be applied for only the most significant bin of the merge index.

[00116] During the process of removal of redundant candidates, comparison between motion vector predictor candidates can also be based on any other information than the motion

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000032

vector values. For example, it can be based on linear or non-linear functions of motion vector values, coding or prediction types of the blocks used to obtain the motion information, block size, the spatial location in the frame/(largest) coding unit/macroblock, the information whether blocks share the same motion with a block, the information whether blocks are in the same coding/prediction unit, etc.

[00117] The following pseudo code illustrates an example embodiment of the invention for constructing the merging list.

[00118] Inputs to this process are

- a luma location (xP, yP) specifying the top-left luma sample of the current prediction unit relative to the top-left sample of the current picture;
- variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH; and
- a variable PartIdx specifying the index of the current prediction unit within the current coding unit.

[00119] Outputs of this process are (with N being replaced by A_0, A_1, B_0, B_1 or B_2 and with X being replaced by 0 or 1)

- the availability flags availableFlagN of the neighbouring prediction units,
- the reference indices refIdxLXN of the neighbouring prediction units,
- the prediction list utilization flags predFlagLXN of the neighbouring prediction units,
- the motion vectors mvLXN of the neighbouring prediction units.

[00120] For the derivation of availableFlagN, with N being A_0, A_1, B_0, B_1 or B_2 and (xN, yN) being (xP−1, yP + nPSH), (xP−1, yP + nPSH−1), (xP + nPSW, yP−1), (xP+nPSW−1, yP−1) or (xP−1, yP−1), the following applies.

- If one of the following conditions is true, the availableFlagN is set equal to 0, both components mvLXN are set equal to 0, refIdxLXN and predFlagLX[xN, yN] of the prediction unit covering luma location (xN, yN) are assigned respectively to mvLXN, refIdxLXN and predFlagLXN.
  - N is equal to B_2 and availableFlagA_0 + availableFlagA_1 + availableFlagB_0 + availableFlagB_1 is equal to 4.
  - The prediction unit covering luma location (xN, yN) is not available or PredMode is MODE_INTRA.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000033

- N is equal to A1 and PartMode of the current prediction unit is PART_Nx2N or PART_nLx2N or PART_nRx2N and PartIdx is equal to 1.
- N is equal to A1 and PartMode of the current prediction unit is PART_2NxN or PART_2NxnU or PART_2NxnD and PartIdx is equal to 1 and the prediction units covering luma location (xP+nPSW−1, yP−1) (N = B1) and luma location (xN, yN) (Cand. N) have identical motion parameters:
  - mvLX[xP+nPSW−1, yP−1] == mvLX[xN, yN]
  - refIdxLX[xP+nPSW−1, yP−1] == refIdxLX[xN, yN]
  - predFlagLX[xP+nPSW−1, yP−1] == predFlagLX[xN, yN]
- N is equal to B1 and PartMode of the current prediction unit is 2NxN or PART_2NxnU or PART_2NxnD and PartIdx is equal to 1.
- N is equal to B1 and the prediction units covering luma location (xP−1, yP+nPSH−1) (N = A1) and luma location (xN, yN) (Cand. N) have identical motion parameters:
  - mvLX[xP−1, yP+nPSH−1] == mvLX[xN, yN]
  - refIdxLX[xP−1, yP+nPSH−1] == refIdxLX[xN, yN]
  - predFlagLX[xP−1, yP+nPSH−1] == predFlagLX[xN, yN]
- N is equal to B0 and the prediction units covering luma location (xP+nPSW−1, yP−1) (N = B1) and luma location (xN, yN) (Cand. N) have identical motion parameters:
  - mvLX[xP+nPSW−1, yP−1] == mvLX[xN, yN]
  - refIdxLX[xP+nPSW−1, yP−1] == refIdxLX[xN, yN]
  - predFlagLX[xP+nPSW−1, yP−1] == predFlagLX[xN, yN]
- N is equal to A0 and the prediction units covering luma location (xP−1, yP+nPSH−1) (N = A1) and luma location (xN, yN) (Cand. N) have identical motion parameters:
  - mvLX[xP−1, yP+nPSH−1] == mvLX[xN, yN]
  - refIdxLX[xP−1, yP+nPSH−1] == refIdxLX[xN, yN]
  - predFlagLX[xP−1, yP+nPSH−1] == predFlagLX[xN, yN]
- N is equal to B2 and the prediction units covering luma location (xP+nPSW−1, yP−1) (N = B1) and luma location (xN, yN) (Cand. N) have identical motion parameters:
  - mvLX[xP+nPSW−1, yP−1] == mvLX[xN, yN]
  - refIdxLX[xP+nPSW−1, yP−1] == refIdxLX[xN, yN]
  - predFlagLX[xP+nPSW−1, yP−1] == predFlagLX[xN, yN]

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000034

- N is equal to B2 and the prediction units covering luma location (xP−1, yP+nPSH−1) (N = A1) and luma location (xN, yN) (Cand. N) have identical motion parameters:
  - mvLX[xP−1, yP+nPSH−1] == mvLX[xN, yN]
  - refIdxLX[xP−1, yP+nPSH−1] == refIdxLX[xN, yN]
  - predFlagLX[xP−1, yP+nPSH−1] == predFlagLX[xN, yN]
- PartMode of the current prediction unit is PART_NxN and PartIdx is equal to 3 and the prediction units covering luma location (xP−1, yP) (PartIdx = 2) and luma location (xP−1, yP−1) (PartIdx = 0) have identical motion parameters:
  - mvLX[xP−1, yP] == mvLX[xP−1, yP−1]
  - refIdxLX[xP−1, yP] == refIdxLX[xP−1, yP−1]
  - predFlagLX[xP−1, yP] == predFlagLX[xP−1, yP−1]

  and the prediction units covering luma location (xP, yP−1) (PartIdx = 1) and luma location (xN, yN) (Cand. N) have identical motion parameters:
  - mvLX[xP, yP−1] == mvLX[xN, yN]
  - refIdxLX[xP, yP−1] == refIdxLX[xN, yN]
  - predFlagLX[xP, yP−1] == predFlagLX[xN, yN]
- PartMode of the current prediction unit is PART_NxN and PartIdx is equal to 3 and the prediction units covering luma location (xP, yP−1) (PartIdx = 1) and luma location (xP−1, yP−1) (PartIdx = 0) have identical motion parameters:
  - mvLX[xP, yP−1] == mvLX[xP−1, yP−1]
  - refIdxLX[xP, yP−1] == refIdxLX[xP−1, yP−1]
  - predFlagLX[xP, yP−1] == predFlagLX[xP−1, yP−1]

  and the prediction units covering luma location (xP−1, yP) (PartIdx = 2) and luma location (xN, yN) (Cand. N) have identical motion parameters:
  - mvLX[xP−1, yP] == mvLX[xN, yN]
  - refIdxLX[xP−1, yP] == refIdxLX[xN, yN]
  - predFlagLX[xP−1, yP] == predFlagLX[xN, yN]
- Otherwise, availableFlagN is set equal to 1 and the variables mvLX[xN, yN], refIdxLX[xN, yN] and predFlagLX[xN, yN] of the prediction unit covering luma location (xN, yN) are assigned respectively to mvLXN, refIdxLXN and predFlagLXN.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000035

[00121] For the motion vector predictor candidate list generation process, each list candidate can include more information than the motion vector value, such as the reference lists used, the reference frames used in each list and motion vector for each list.

[00122] When all motion vector candidates have been examined, one motion vector is selected to be used as the motion vector for the current block. The motion vector selector 364 may examine different motion vectors in the list and determine which motion vector provides the most efficient encoding result, or the selection of the motion vector may be based on to other criteria as well. Information of the selected motion vector is provided for the mode selector for encoding and transmission to the decoder or for storage when the mode selector determines to use inter prediction for the current block. The information may include the index of the motion vector in the list, and/or motion vector parameters or other appropriate information.

[00123] The selected motion vector and the block relating to the motion vector is used to generate the prediction representation of the image block 312 which is provided as the output of the mode selector. The output may be used by the first summing device 321 to produce the first prediction error signal 320, as was described above.

[00124] The selected motion vector predictor candidate can be modified by adding a motion vector difference or can be used directly as the motion vector of the block. Moreover, after the motion compensation is performed by using the selected motion vector predictor candidate, the residual signal of the block can be transform coded or skipped to be coded.

[00125] Although the embodiments above have been described with respect to the size of the macroblock being 16x16 pixels, it would be appreciated that the methods and apparatus described may be configured to handle macroblocks of different pixel sizes.

[00126] In the following the operation of an example embodiment of the decoder 600 is depicted in more detail with reference to Figure 7.

[00127] At the decoder side similar operations are performed to reconstruct the image blocks. Figure 7 shows a block diagram of a video decoder 700 suitable for employing embodiments of the invention and Figures 8a and 8b show a flow diagram of an example of a method in the video decoder. The bitstream to be decoded may be received from the encoder, from a network element, from a storage medium or from another source. The decoder is aware of the structure of the bitstream so that it can determine the meaning of the entropy coded codewords and may decode the bitstream by an entropy decoder 701 which performs entropy

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000036

decoding on the received signal. The entropy decoder thus performs the inverse operation to the entropy encoder 330 of the encoder described above. The entropy decoder 701 outputs the results of the entropy decoding to a prediction error decoder 702 and a pixel predictor 704.

[00128] In some embodiments the entropy coding may not be used but another channel encoding may be in use, or the encoded bitstream may be provided to the decoder 700 without channel encoding. The decoder 700 may comprise a corresponding channel decoder to obtain the encoded codewords from the received signal.

[00129] The pixel predictor 704 receives the output of the entropy decoder 701. The output of the entropy decoder 701 may include an indication on the prediction mode used in encoding the current block. A predictor selector 714 within the pixel predictor 704 determines that an intra-prediction or an inter-prediction is to be carried out. The predictor selector 714 may furthermore output a predicted representation of an image block 716 to a first combiner 713. The predicted representation of the image block 716 is used in conjunction with the reconstructed prediction error signal 712 to generate a preliminary reconstructed image 718. The preliminary reconstructed image 718 may be used in the predictor 714 or may be passed to a filter 720. The filter 720, if used, applies a filtering which outputs a final reconstructed signal 722. The final reconstructed signal 722 may be stored in a reference frame memory 724, the reference frame memory 724 further being connected to the predictor 714 for prediction operations.

[00130] Also the prediction error decoder 702 receives the output of the entropy decoder 701. A dequantizer 792 of the prediction error decoder 702 may dequantize the output of the entropy decoder 701 and the inverse transform block 793 may perform an inverse transform operation to the dequantized signal output by the dequantizer 792. The output of the entropy decoder 701 may also indicate that prediction error signal is not to be applied and in this case the prediction error decoder produces an all zero output signal.

[00131] The decoder selects the 16x16 pixel residual macroblock to reconstruct. This residual macroblock is also called as a current block.

[00132] The decoder may receive information on the encoding mode used in encoding of the current block. The indication is decoded, when necessary, and provided to the reconstruction processor 791 of the prediction selector 714. The reconstruction processor 791 examines the indication and selects one of the intra-prediction mode(s), if the indication indicates that the

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000037

block has been encoded using intra-prediction, or the inter-prediction mode, if the indication indicates that the block has been encoded using inter-prediction.

[00133] For inter-prediction mode the reconstruction processor 791 may comprise one or more elements corresponding to the prediction processor 362 of the encoder, such as a motion vector definer, a prediction list modifier and/or a motion vector selector.

[00134] The reconstruction processor 791 initializes a motion vector prediction list to default values in block 800. As was the case in the encoding part, in this example the spatial motion prediction candidates are the spatial neighbour blocks A0, A1, B0, B1, B2 and these spatial motion prediction candidates are processed in the same predetermined order than in the encoder: A1, B1, B0, A0 and B2. The first spatial motion prediction candidate to be selected for further examination is thus A1. Before further examination is performed for the selected spatial motion prediction candidate, it is examined whether the merge list already contains a maximum number of spatial motion prediction candidates. If the number of spatial motion prediction candidates in the merge list is not less than the maximum number, the selected spatial motion prediction candidate is not included in the merge list and the process of constructing the merge list can be stopped 826. On the other hand, if the number of spatial motion prediction candidates in the merge list is less than the maximum number, a further analyses of the selected spatial motion prediction candidate is performed (blocks 804-822).

[00135] The decoder examines 804 if the prediction unit or block covering the spatial motion prediction candidate block is not available for motion prediction. If so, the candidate is not included in the merge list. The reason that the block is not available may be that the block is either coded in intra mode or resides in a different slice or outside of the picture area.

[00136] In addition to the common conditions above, for each spatial motion prediction candidate, if any of the following conditions holds, then the candidate is not included in the merge list, otherwise, it is included.

[00137] The decoder determines 806 which spatial motion prediction candidate of the set of spatial motion prediction candidates is in question. If the spatial motion prediction candidate is the block A1, one or more of the following conditions may be examined 808, 810 to determine whether to include this spatial motion prediction candidate in the merge list or not. If the current coding unit 100 is vertically split into two rectangle prediction units 103, 104 as depicted in Figure 10b and the current prediction unit is the second prediction unit 104 in the

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000038

coding/decoding order (808), this spatial motion prediction candidate is not included in the merge list. If the current coding unit 100 is not vertically split into two rectangle prediction units but it is horizontally split into two rectangle prediction units 101, 102 as depicted in Figure 10a and the current prediction unit is the second prediction unit in the coding/decoding order and the block A1 has the same motion information as the block B1 (810), this spatial motion prediction candidate (block A1) is not included in the merge list. In the example of Figure 10a the second prediction unit is the lower prediction unit 102 of the coding unit 100 and in the example of Figure 10b the second prediction unit is the rightmost prediction unit 104 of the coding unit 100. If none of the conditions above is fulfilled the block A1 is included in the merge list as a spatial motion prediction candidate (824).

[00138] If the spatial motion prediction candidate is the block B1, one or more of the following conditions may be examined 812, 814 to determine whether to include this spatial motion prediction candidate in the merge list or not. If the current coding unit 100 is horizontally split into two rectangle prediction units 101, 102 as depicted in Figure 10a and the current prediction unit is the second prediction unit 104 in the coding/decoding order (812), this spatial motion prediction candidate is not included in the merge list. If the current coding unit 100 is not horizontally split into two rectangle prediction units and if the block B1 has the same motion information than the block A1 (814), this spatial motion prediction candidate (block B1) is not included in the merge list. If none of the conditions above is fulfilled the block B1 is included in the merge list as a spatial motion prediction candidate (824).

[00139] If the spatial motion prediction candidate is the block B0, this spatial motion prediction candidate is not included in the merge list if the block B0 has the same motion information than the block B1 (816). Otherwise, if the number of spatial motion prediction candidates in the merge list is less than the maximum number of spatial motion prediction candidates, this spatial motion prediction candidate (block B0) is included in the merge list (824).

[00140] If the spatial motion prediction candidate is the block A0, this spatial motion prediction candidate is not included in the merge list if the block A0 has the same motion information than the block A1 (818). Otherwise, if the number of spatial motion prediction candidates in the merge list is less than the maximum number of spatial motion prediction

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000039

candidates, this spatial motion prediction candidate (block A0) is included in the merge list (824).

[00141] If the spatial motion prediction candidate is the block B2, this spatial motion prediction candidate is not included in the merge list if the maximum number of spatial motion prediction candidates is four and the other blocks A0, A1, B0, and B1 are all decided to be included in the merge list (820). Otherwise, if the number of spatial motion prediction candidates in the merge list is less than the maximum number of spatial motion prediction candidates, the block B2 is not included in the merge list if the block B2 has the same motion information than the block B1 or the block A1 (822).

[00142] Then, after processing the blocks A1, B1, B0, A0 and B2 and including a subset of them in the merge list based on the above described conditions, no more redundancy check between these candidates are performed and remaining temporal motion prediction candidate and/or other possible additional candidates may be processed.

[00143] When the merge list has been constructed the decoder may use 828 the indication of the motion vector received from the encoder to select the motion vector for decoding the current block. The indication may be, for example, an index to the merge list.

[00144] Basically, after the reconstruction processor 791 has constructed the merge list, it would correspond with the merge list constructed by the encoder if the reconstruction processor 791 has the same information available than the encoder had. If some information has been lost during transmission the information from the encoder to the decoder, it may affect the generation of the merge list in the decoder 700.

[00145] The above examples describe the operation mainly in the merge mode but the encoder and decoder may also operate in other modes.

[00146] The embodiments of the invention described above describe the codec in terms of separate encoder and decoder apparatus in order to assist the understanding of the processes involved. However, it would be appreciated that the apparatus, structures and operations may be implemented as a single encoder-decoder apparatus/structure/operation. Furthermore in some embodiments of the invention the coder and decoder may share some or all common elements.

[00147] Although the above examples describe embodiments of the invention operating within a codec within an electronic device, it would be appreciated that the invention as described below may be implemented as part of any video codec. Thus, for example,

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000040

embodiments of the invention may be implemented in a video codec which may implement video coding over fixed or wired communication paths.

[00148] Thus, user equipment may comprise a video codec such as those described in embodiments of the invention above.

[00149] It shall be appreciated that the term user equipment is intended to cover any suitable type of wireless user equipment, such as mobile telephones, portable data processing devices or portable web browsers.

[00150] Furthermore elements of a public land mobile network (PLMN) may also comprise video codecs as described above.

[00151] In general, the various embodiments of the invention may be implemented in hardware or special purpose circuits, software, logic or any combination thereof. For example, some aspects may be implemented in hardware, while other aspects may be implemented in firmware or software which may be executed by a controller, microprocessor or other computing device, although the invention is not limited thereto. While various aspects of the invention may be illustrated and described as block diagrams, flow charts, or using some other pictorial representation, it is well understood that these blocks, apparatus, systems, techniques or methods described herein may be implemented in, as non-limiting examples, hardware, software, firmware, special purpose circuits or logic, general purpose hardware or controller or other computing devices, or some combination thereof.

[00152] The embodiments of this invention may be implemented by computer software executable by a data processor of the mobile device, such as in the processor entity, or by hardware, or by a combination of software and hardware. Further in this regard it should be noted that any blocks of the logic flow as in the Figures may represent program steps, or interconnected logic circuits, blocks and functions, or a combination of program steps and logic circuits, blocks and functions. The software may be stored on such physical media as memory chips, or memory blocks implemented within the processor, magnetic media such as hard disk or floppy disks, and optical media such as for example DVD and the data variants thereof, CD.

[00153] The memory may be of any type suitable to the local technical environment and may be implemented using any suitable data storage technology, such as semiconductor based memory devices, magnetic memory devices and systems, optical memory devices and systems, fixed memory and removable memory. The data processors may be of any type suitable to the

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000041

local technical environment, and may include one or more of general purpose computers, special purpose computers, microprocessors, digital signal processors (DSPs) and processors based on multi core processor architecture, as non limiting examples.

[00154] Embodiments of the inventions may be practiced in various components such as integrated circuit modules. The design of integrated circuits is by and large a highly automated process. Complex and powerful software tools are available for converting a logic level design into a semiconductor circuit design ready to be etched and formed on a semiconductor substrate.

[00155] Programs, such as those provided by Synopsys, Inc. of Mountain View, California and Cadence Design, of San Jose, California automatically route conductors and locate components on a semiconductor chip using well established rules of design as well as libraries of pre stored design modules. Once the design for a semiconductor circuit has been completed, the resultant design, in a standardized electronic format (e.g., Opus, GDSII, or the like) may be transmitted to a semiconductor fabrication facility or "fab" for fabrication.

[00156] The foregoing description has provided by way of exemplary and non-limiting examples a full and informative description of the exemplary embodiment of this invention. However, various modifications and adaptations may become apparent to those skilled in the relevant arts in view of the foregoing description, when read in conjunction with the accompanying drawings and the appended claims. However, all such and similar modifications of the teachings of this invention will still fall within the scope of this invention.

[00157] In the following some examples will be provided.

[00158] In some embodiments a method comprises:

receiving a block of pixels including a prediction unit; determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000042

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[00159] In some embodiments the method comprises including neighbouring blocks of the received block of pixels in the set of spatial motion vector prediction candidates.

[00160] In some embodiments the method comprises constructing the set of spatial motion vector predictions by using motion vectors of one or more encoded blocks in a same frame than the block of pixels.

[00161] In some embodiments the method comprises selecting spatial motion vector prediction candidates from the set of spatial motion vector prediction candidates as the potential spatial motion vector prediction candidate in a predetermined order.

[00162] In some embodiments the method comprises comparing motion information of the potential spatial motion vector prediction candidate with motion information of at most one other spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates.

[00163] In some embodiments the method comprises prediction unit and a second prediction unit; and if so, excluding the potential spatial motion vector prediction candidate from the merge list if the prediction unit is the second prediction unit.

[00164] In some embodiments the method comprises

determining a maximum number of spatial motion vector prediction candidates to be included in a merge list; and

limiting the number of spatial motion vector prediction candidates in the merge list smaller or equal to the maximum number .

[00165] In some embodiments the method comprises

examining, if the number of spatial motion vector prediction candidates in the merge list smaller than the maximum number;

if so, examining whether a prediction unit to which the potential spatial motion vector prediction candidate belongs is available for motion prediction;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000043

if so, performing at least one of the following:

for the potential spatial motion vector prediction candidate on the left side of the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- the received block of pixels is vertically divided into a first prediction unit and a second prediction unit, and the prediction unit is the second prediction unit;
- the received block of pixels is horizontally divided into a first prediction unit and a second prediction unit, and if the prediction unit is the second prediction unit, and the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;

for the potential spatial motion vector prediction candidate above the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- the received block of pixels is horizontally divided into a first prediction unit and a second prediction unit, and the prediction unit is the second prediction unit;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit;

for the potential spatial motion vector prediction candidate, which is on the right side of the potential spatial motion vector prediction candidate above the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;

for the potential spatial motion vector prediction candidate, which is below the potential spatial motion vector prediction candidate on the left side of the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000044

for the potential spatial motion vector prediction candidate cornerwise neighbouring the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- all the other potential spatial motion vector prediction candidates have been included in the merge list;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit.

[00166] In some embodiments the method comprises including a temporal motion prediction candidate into the merge list.

[00167] In some embodiments the method comprises selecting one motion vector prediction candidate from the merge list to represent a motion vector prediction for the block of pixels.

[00168] In some embodiments a method according to the second aspect comprises:

receiving an encoded block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of another spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000045

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[00169] In some embodiments the method comprises including neighbouring blocks of the received encoded block of pixels in the set of spatial motion vector prediction candidates.

[00170] In some embodiments the method comprises constructing the set of spatial motion vector predictions by using motion vectors of one or more decoded blocks in a same frame than the received encoded block of pixels.

[00171] In some embodiments the method comprises selecting spatial motion vector prediction candidates from the set of spatial motion vector prediction candidates as the potential spatial motion vector prediction candidate in a predetermined order.

[00172] In some embodiments the method comprises comparing motion information of the potential spatial motion vector prediction candidate with motion information of at most one other spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates.

[00173] In some embodiments the method comprises examining whether the received encoded block of pixels is divided into a first prediction unit and a second prediction unit; and if so, excluding the potential spatial motion vector prediction candidate from the merge list if the prediction unit is the second prediction unit.

[00174] In some embodiments the method comprises

determining a maximum number of spatial motion vector prediction candidates to be included in a merge list; and

limiting the number of spatial motion vector prediction candidates in the merge list smaller or equal to the maximum number.

[00175] In some embodiments the method comprises

examining, if the number of spatial motion vector prediction candidates in the merge list smaller than the maximum number;

if so, examining whether a prediction unit to which the potential spatial motion vector prediction candidate belongs is available for motion prediction;

if so, performing at least one of the following:

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000046

for the potential spatial motion vector prediction candidate on the left side of the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- the received encoded block of pixels is vertically divided into a first prediction unit and a second prediction unit, and the prediction unit is the second prediction unit;
- the received encoded block of pixels is horizontally divided into a first prediction unit and a second prediction unit, and if the prediction unit is the second prediction unit, and the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;

for the potential spatial motion vector prediction candidate above the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- the received encoded block of pixels is horizontally divided into a first prediction unit and a second prediction unit, and the prediction unit is the second prediction unit;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit;

for the potential spatial motion vector prediction candidate, which is on the right side of the potential spatial motion vector prediction candidate above the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;

for the potential spatial motion vector prediction candidate, which is below the potential spatial motion vector prediction candidate on the left side of the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000047

for the potential spatial motion vector prediction candidate cornerwise neighbouring the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- all the other potential spatial motion vector prediction candidates have been included in the merge list;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit.

[00176] In some embodiments the method comprises including a temporal motion prediction candidate into the merge list.

[00177] In some embodiments the method comprises selecting one motion vector prediction candidate from the merge list to represent a motion vector prediction for the received encoded block of pixels.

[00178] In some embodiments an apparatus according to the third aspect comprises a processor and a memory including computer program code, the memory and the computer program code configured to, with the processor, cause the apparatus to:

receive a block of pixels

including a prediction unit;

determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000048

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[00179] In some embodiments an apparatus according to the fourth aspect comprises a processor and a memory including computer program code, the memory and the computer program code configured to, with the processor, cause the apparatus to:

receive an encoded block of pixels including a prediction unit;

determine a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

select a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determine a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

compare motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

exclude the first spatial motion vector prediction candidate from the merge list, if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other

[00180] In some embodiments a storage medium having stored thereon a computer program code a computer executable program code for use by an encoder, said program codes comprise instructions for use by an encoder, said program code comprises instructions for:

receiving a block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000049

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[00181] In some embodiments a storage medium having stored thereon a computer program code a computer executable program code for use by an encoder, said program codes comprise instructions for use by an encoder, said program code comprises instructions for:

receiving an encoded block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

[00182] In some embodiments an apparatus comprises:

means for receiving a block of pixels including a prediction unit;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000050

means for selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

means for determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

means for comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

means for excluding the first spatial motion vector prediction candidate from the merge list, if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other.

[00183] In some embodiments an apparatus comprises:

means for receiving an encoded block of pixels including a prediction unit;

means for determining a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

means for selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

means for determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

means for comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

means for excluding the first spatial motion vector prediction candidate from the merge list, if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000051

Claims:

1. A method comprising:

receiving a block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

2. The method according to claim 1 comprising selecting spatial motion vector prediction candidates from the set of spatial motion vector prediction candidates as the potential spatial motion vector prediction candidate in a predetermined order.

3. The method according to claim 1 , comprising comparing motion information of the potential spatial motion vector prediction candidate with motion information of at most one other spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates.

4. The method according to claim 1 comprising examining whether the received block of pixels is divided into a first prediction unit and a second prediction unit; and if so, excluding the potential spatial motion vector prediction candidate from the merge list if the prediction unit is the second prediction unit.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000052

5. The method according to claim 1, further comprising

determining a maximum number of spatial motion vector prediction candidates to be included in a merge list; and

limiting the number of spatial motion vector prediction candidates in the merge list smaller or equal to the maximum number .

6. The method according to claim 1 comprising:

examining, if the number of spatial motion vector prediction candidates in the merge list smaller than the maximum number;

if so, examining whether a prediction unit to which the potential spatial motion vector prediction candidate belongs is available for motion prediction;

if so, performing at least one of the following:

for the potential spatial motion vector prediction candidate on the left side of the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- the received block of pixels is vertically divided into a first prediction unit and a second prediction unit, and the prediction unit is the second prediction unit;
- the received block of pixels is horizontally divided into a first prediction unit and a second prediction unit, and if the prediction unit is the second prediction unit, and the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;

for the potential spatial motion vector prediction candidate above the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- the received block of pixels is horizontally divided into a first prediction unit and a second prediction unit, and the prediction unit is the second prediction unit;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000053

- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit;

for the potential spatial motion vector prediction candidate, which is on the right side of the potential spatial motion vector prediction candidate above the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;

for the potential spatial motion vector prediction candidate, which is below the potential spatial motion vector prediction candidate on the left side of the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit;

for the potential spatial motion vector prediction candidate cornerwise neighbouring the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- all the other potential spatial motion vector prediction candidates have been included in the merge list;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit.

7. The method according to claim 1 further comprising including a temporal motion prediction candidate into the merge list.

8. The method according to claim 1 comprising selecting one motion vector prediction candidate from the merge list to represent a motion vector prediction for the block of pixels.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000054

9. A method comprising:

receiving an encoded block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of another spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

10. The method according to claim 9 comprising comparing motion information of the potential spatial motion vector prediction candidate with motion information of at most one other spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates.

11. The method according to claim 9 comprising examining whether the received encoded block of pixels is divided into a first prediction unit and a second prediction unit; and if so, excluding the potential spatial motion vector prediction candidate from the merge list if the prediction unit is the second prediction unit.

12. The method according to claim 9 further comprising

determining a maximum number of spatial motion vector prediction candidates to be included in a merge list; and

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000055

limiting the number of spatial motion vector prediction candidates in the merge list smaller or equal to the maximum number.

13. The method according to claim 9 comprising:

examining, if the number of spatial motion vector prediction candidates in the merge list smaller than the maximum number;

if so, examining whether a prediction unit to which the potential spatial motion vector prediction candidate belongs is available for motion prediction;

if so, performing at least one of the following:

for the potential spatial motion vector prediction candidate on the left side of the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- the received encoded block of pixels is vertically divided into a first prediction unit and a second prediction unit, and the prediction unit is the second prediction unit;
- the received encoded block of pixels is horizontally divided into a first prediction unit and a second prediction unit, and if the prediction unit is the second prediction unit, and the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;

for the potential spatial motion vector prediction candidate above the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- the received encoded block of pixels is horizontally divided into a first prediction unit and a second prediction unit, and the prediction unit is the second prediction unit;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit;

for the potential spatial motion vector prediction candidate, which is on the right side of the potential spatial motion vector prediction candidate above the prediction unit, excluding the

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000056

potential spatial motion vector prediction candidate from the merge list if the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;

for the potential spatial motion vector prediction candidate, which is below the potential spatial motion vector prediction candidate on the left side of the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit;

for the potential spatial motion vector prediction candidate cornerwise neighbouring the prediction unit, excluding the potential spatial motion vector prediction candidate from the merge list if any of the following conditions are fulfilled:

- all the other potential spatial motion vector prediction candidates have been included in the merge list;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate above the prediction unit;
- the potential spatial motion vector prediction candidate has essentially similar motion information than the spatial motion vector prediction candidate on the left side of the prediction unit.

14. The method according to claim 9 comprising selecting one motion vector prediction candidate from the merge list to represent a motion vector prediction for the received encoded block of pixels.

15. An apparatus comprising a processor and a memory including computer program code, the memory and the computer program code configured to, with the processor, cause the apparatus to:

receiving a block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000057

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

16. An apparatus comprising a processor and a memory including computer program code, the memory and the computer program code configured to, with the processor, cause the apparatus to:

receive an encoded block of pixels including a prediction unit;

determine a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

select a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determine a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

compare motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

exclude the first spatial motion vector prediction candidate from the merge list, if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000058

17. A storage medium having stored thereon a computer executable program code for use by an encoder, said program codes comprise instructions for use by an encoder, said program code comprises instructions for:

receiving a block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

18. A storage medium having stored thereon a computer executable program code for use by an encoder, said program codes comprise instructions for use by an encoder, said program code comprises instructions for:

receiving an encoded block of pixels including a prediction unit;

determining a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000059

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

19. An apparatus comprising:

means for receiving a block of pixels including a prediction unit;

means for determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;

means for selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

means for determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

means for comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

means for excluding the first spatial motion vector prediction candidate from the merge list, if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other.

20. An apparatus comprising:

means for receiving an encoded block of pixels including a prediction unit;

means for determining a set of spatial motion vector prediction candidates for the encoded block of pixels; the spatial motion vector prediction candidates being provided with motion information;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000060

means for selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

means for determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

means for comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

means for excluding the first spatial motion vector prediction candidate from the merge list, if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000061

ABSTRACT

The invention relates to a method for encoding, a method for decoding, an apparatus, computer program products, an encoder and a decoder for video information. The motion vector for a block in a video image is predicted from a set of motion vector prediction candidates determined based on previously-coded motion vectors. A motion vector prediction candidate is included in the set based on the location of the block associated with the first spatial motion vector prediction candidate and in comparison with motion vector prediction candidates already in the set.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000062

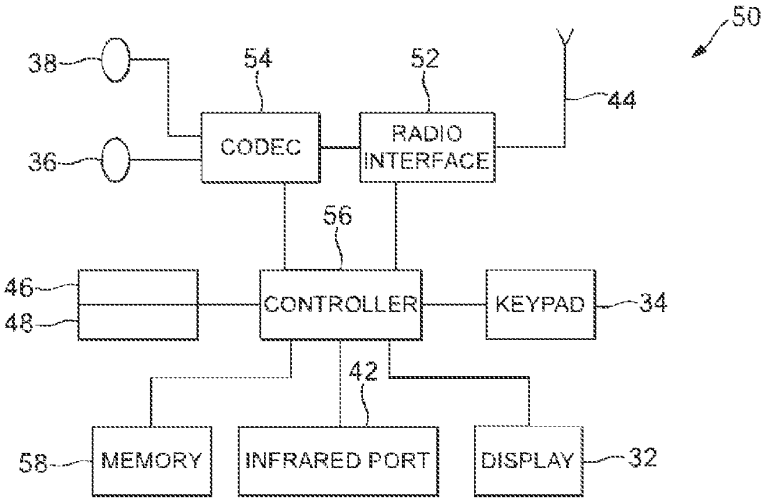

50
38
54
52
44
CODEC
RADIO INTERFACE
36
56
46
48
CONTROLLER
KEYPAD
34
42
58
MEMORY
INFRARED PORT
DISPLAY
32

FIG.1

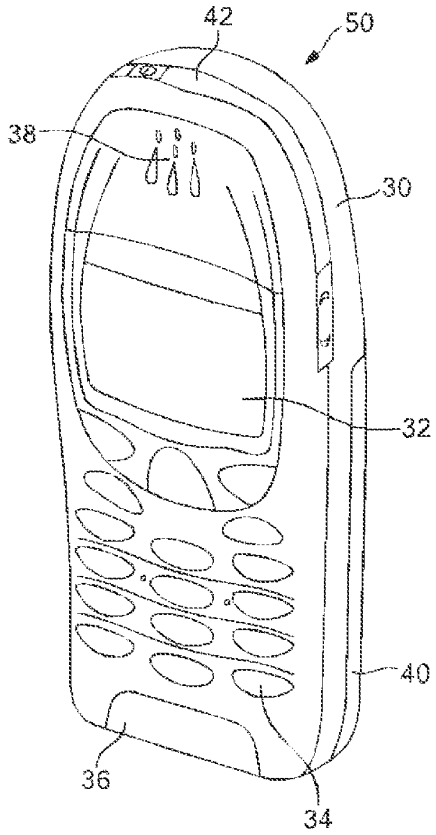

42
50
38
30
32
40
36
34

FIG. 2

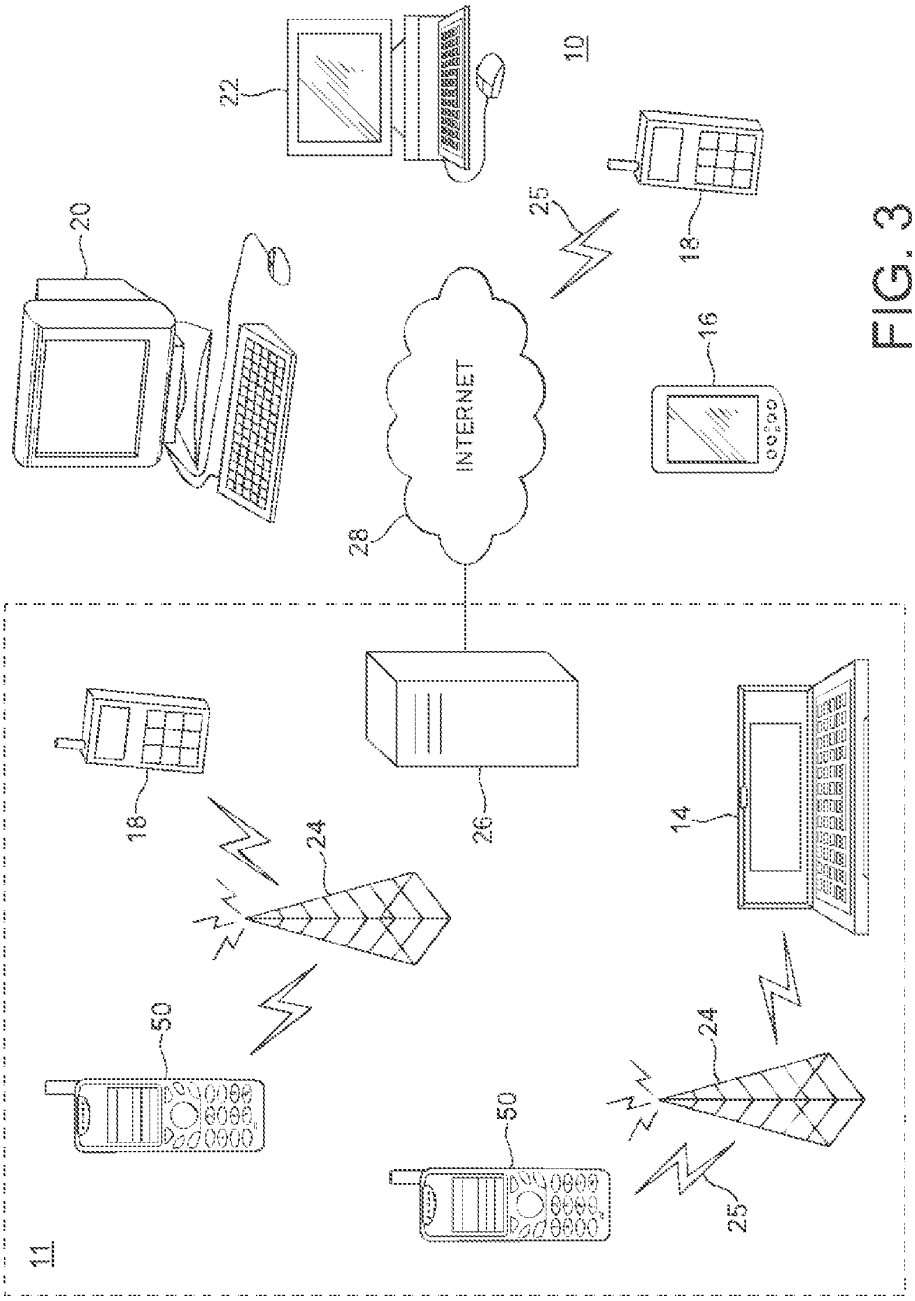


FIG. 3

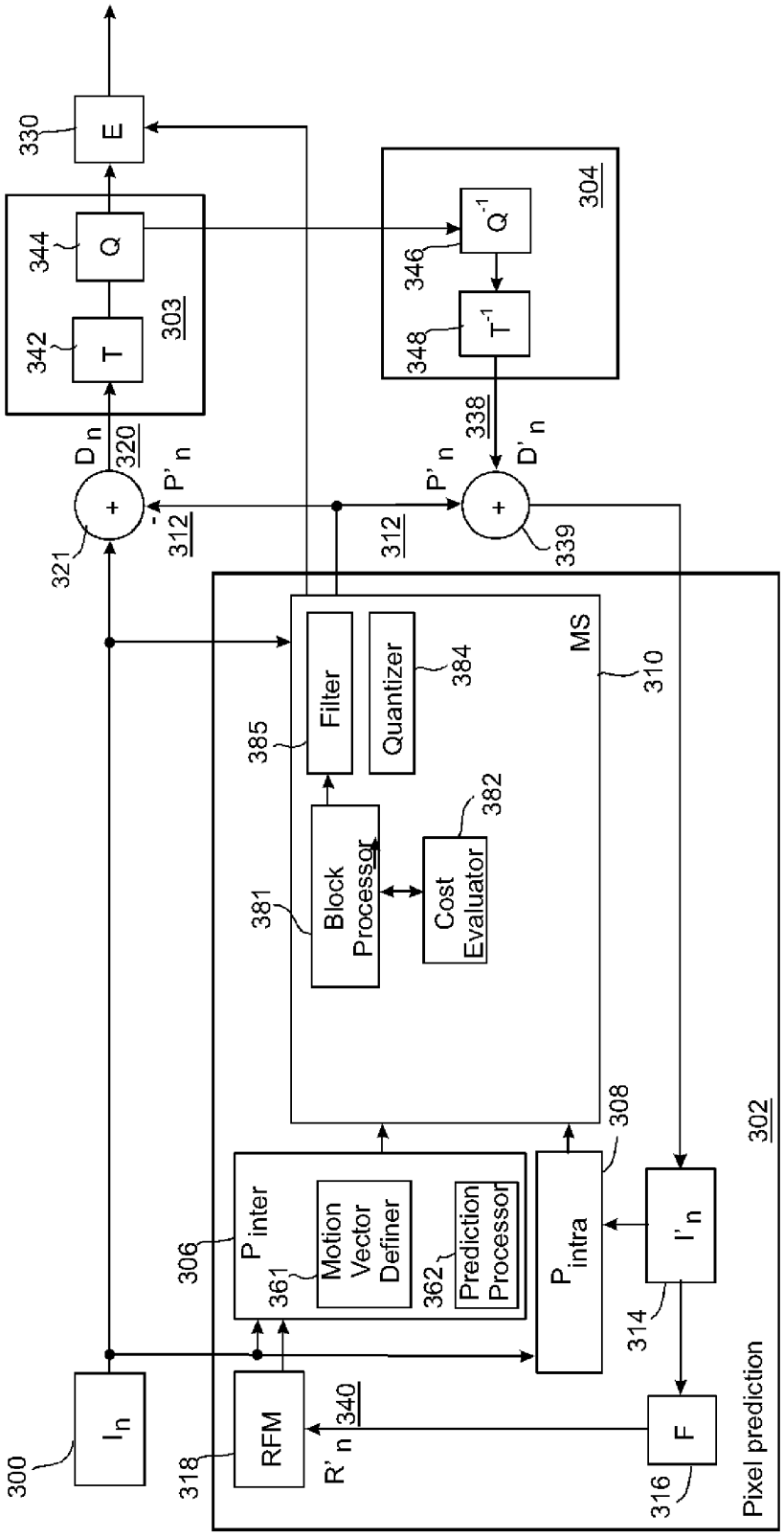


Fig. 4a

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000065

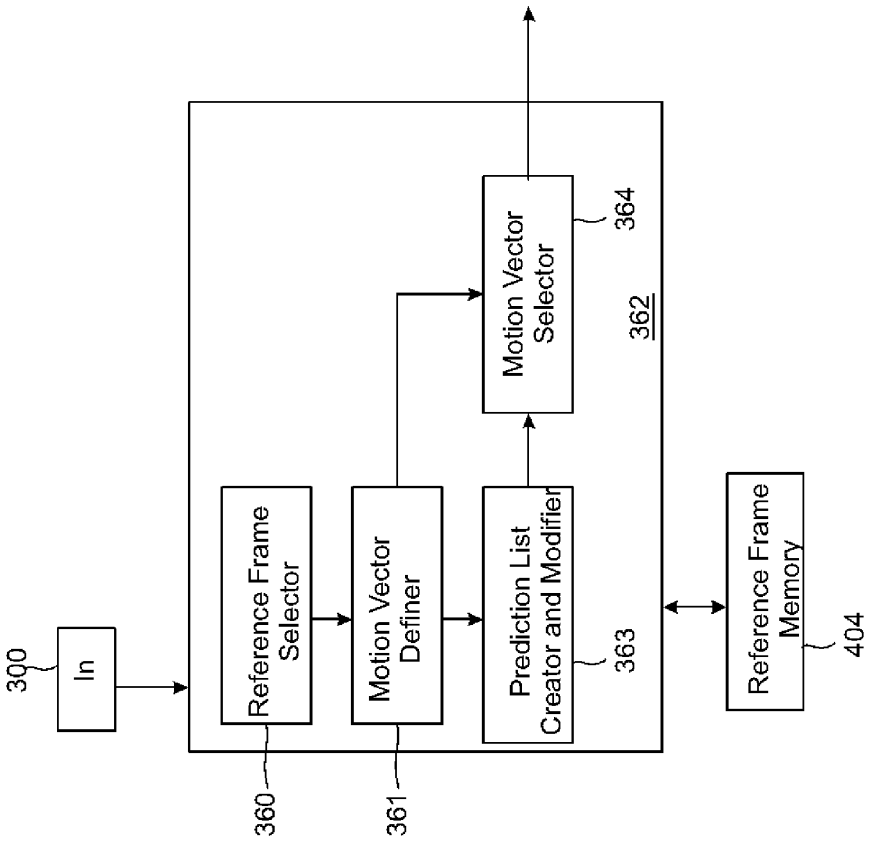

300
In
Reference Frame Selector
Motion Vector Definer
Prediction List Creator and Modifier
363
Motion Vector Selector
364
362
Reference Frame Memory
404
360
361

Fig. 4b

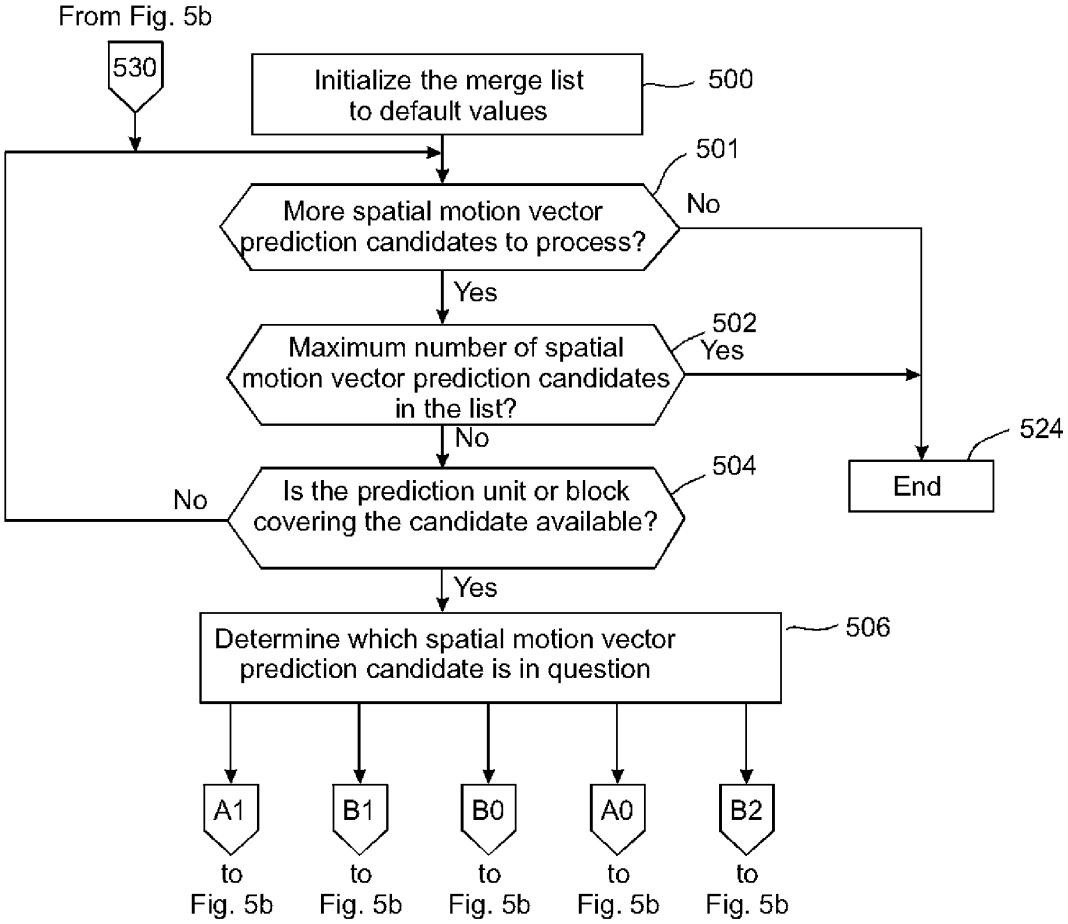

From Fig. 5b
530
Initialize the merge list to default values
500
501
More spatial motion vector prediction candidates to process?
No
Yes
502
Maximum number of spatial motion vector prediction candidates in the list?
Yes
No
524
End
504
Is the prediction unit or block covering the candidate available?
No
Yes
506
Determine which spatial motion vector prediction candidate is in question
A1
B1
B0
A0
B2
to Fig. 5b
to Fig. 5b
to Fig. 5b
to Fig. 5b
to Fig. 5b

Fig. 5a

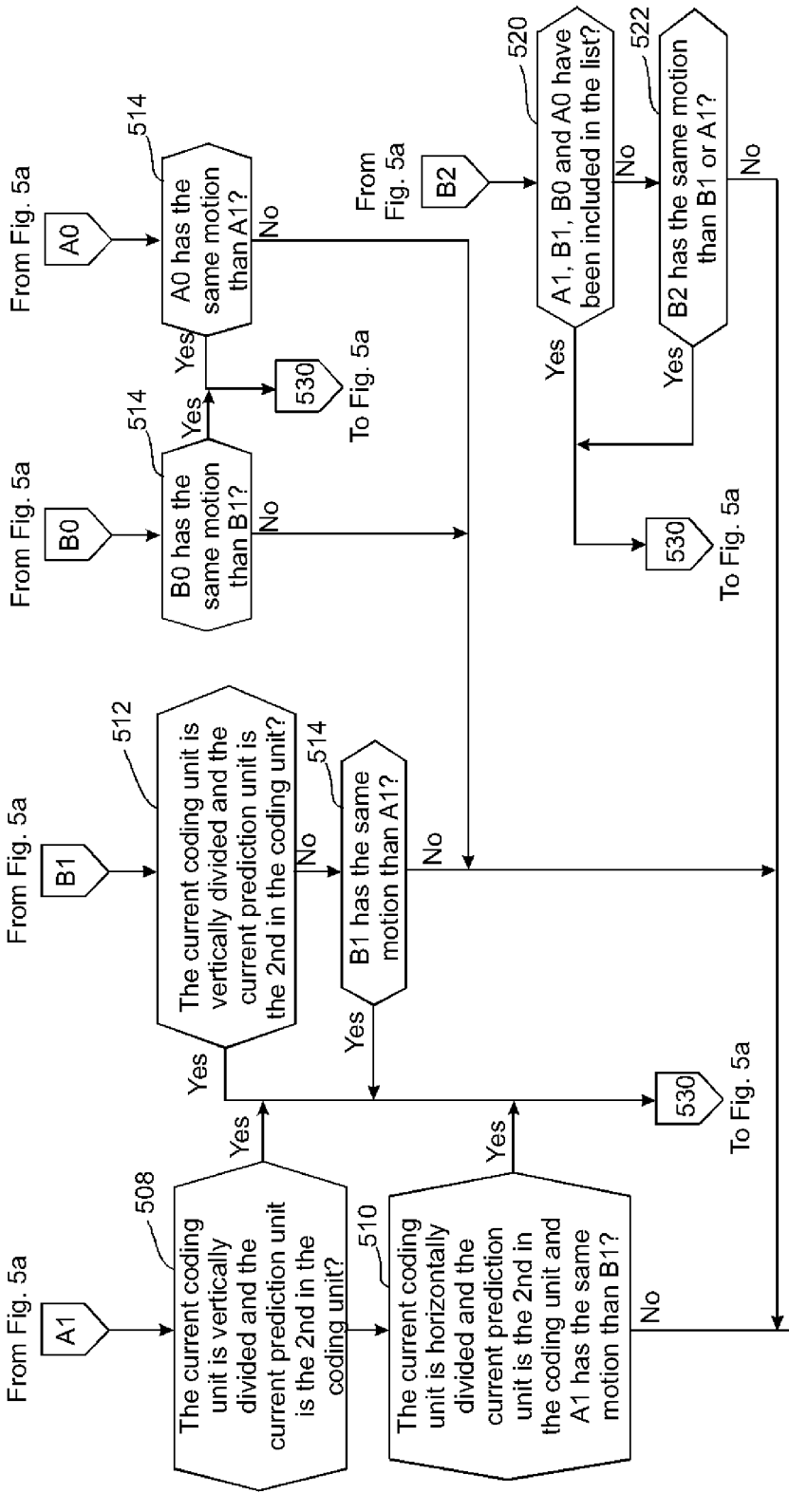

From Fig. 5a
A1
The current coding unit is vertically divided and the current prediction unit is the 2nd in the coding unit?
508
Yes
The current coding unit is horizontally divided and the current prediction unit is the 2nd in the coding unit and A1 has the same motion than B1?
510
Yes
No
From Fig. 5a
B1
The current coding unit is vertically divided and the current prediction unit is the 2nd in the coding unit?
512
Yes
No
B1 has the same motion than A1?
514
Yes
No
From Fig. 5a
B0
B0 has the same motion than B1?
514
Yes
No
From Fig. 5a
A0
A0 has the same motion than A1?
514
Yes
No
From Fig. 5a
B2
A1, B1, B0 and A0 have been included in the list?
520
Yes
No
B2 has the same motion than B1 or A1?
522
Yes
No
530
To Fig. 5a
530
To Fig. 5a
530
To Fig. 5a
530
To Fig. 5a
Insert the candidate into the list
510
530
To Fig. 5a

Fig. 5b

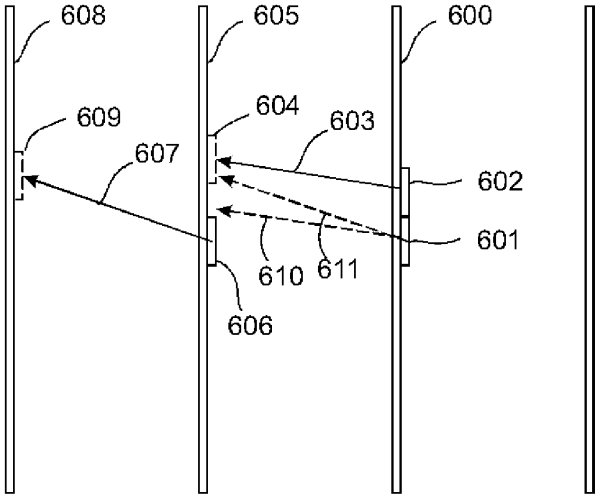

608
605
600
609
604
603
607
602
601
610
611
606

Fig. 6a

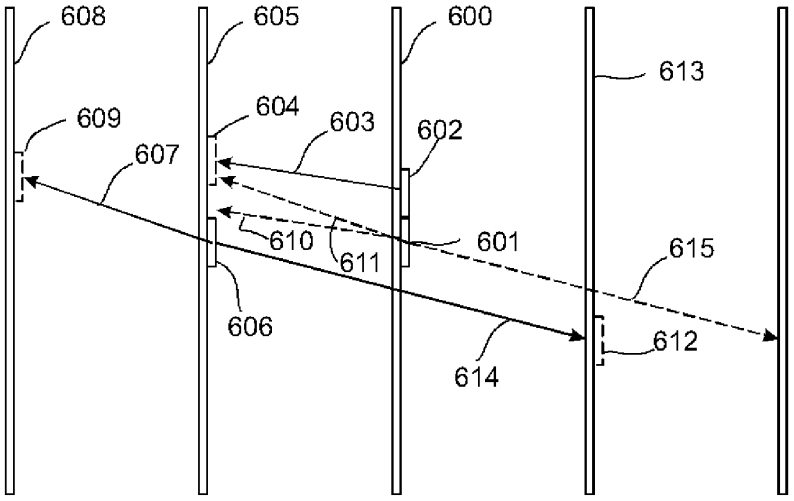

608
605
600
613
609
604
603
602
607
610
611
601
615
606
612
614

Fig. 6b

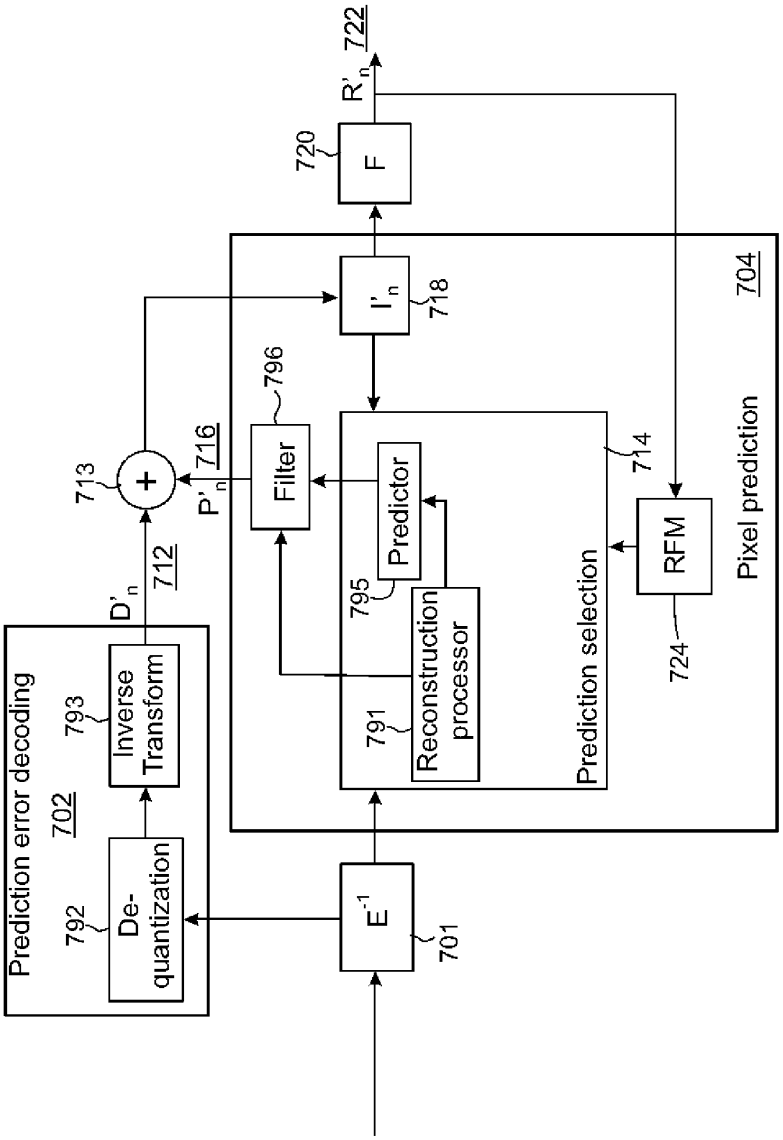

Prediction error decoding
702
792
De-quantization
793
Inverse Transform
D'n
712
713
+
P'n
716
Filter
796
Predictor
795
Reconstruction processor
791
Prediction selection
714
RFM
724
Pixel prediction
704
I'n
718
720
F
R'n
722
E-1
701

Fig. 7

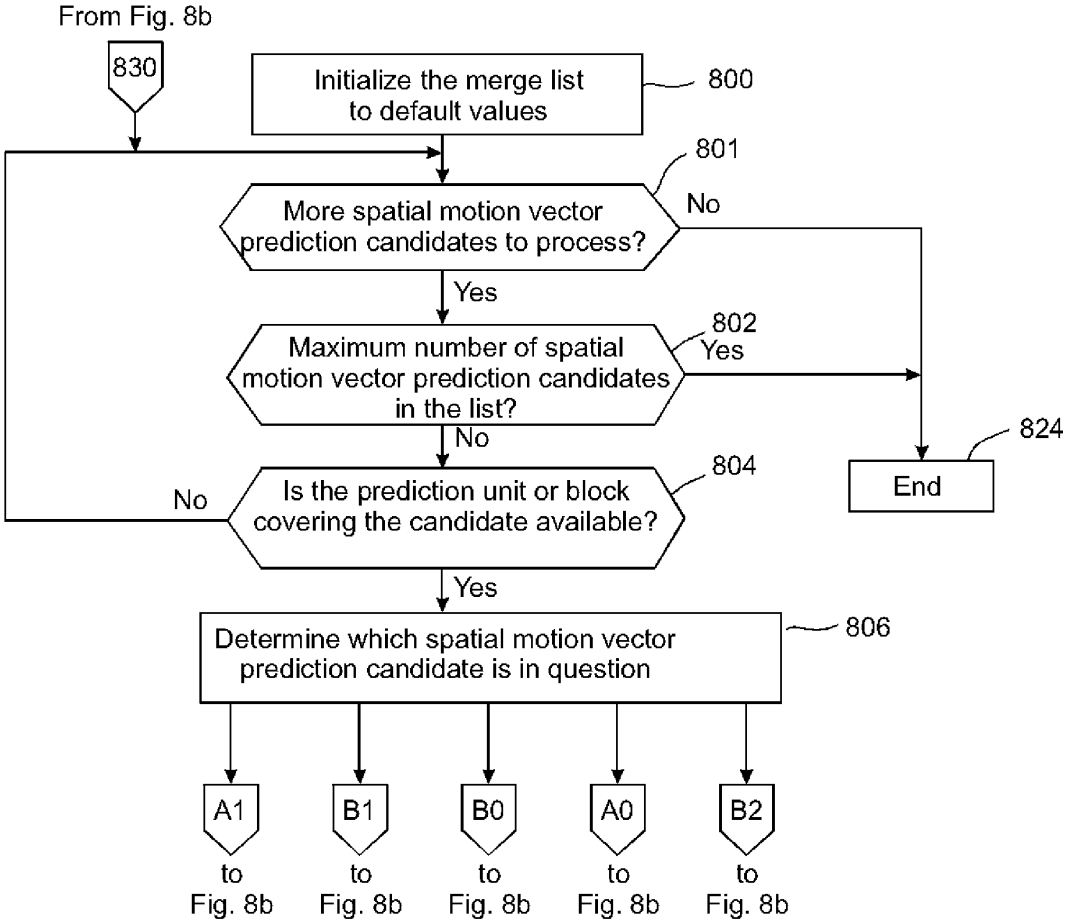

From Fig. 8b
830
Initialize the merge list to default values
800
801
More spatial motion vector prediction candidates to process?
No
Yes
802
Maximum number of spatial motion vector prediction candidates in the list?
Yes
No
824
End
804
Is the prediction unit or block covering the candidate available?
No
Yes
806
Determine which spatial motion vector prediction candidate is in question
A1
B1
B0
A0
B2
to Fig. 8b
to Fig. 8b
to Fig. 8b
to Fig. 8b
to Fig. 8b

Fig. 8a

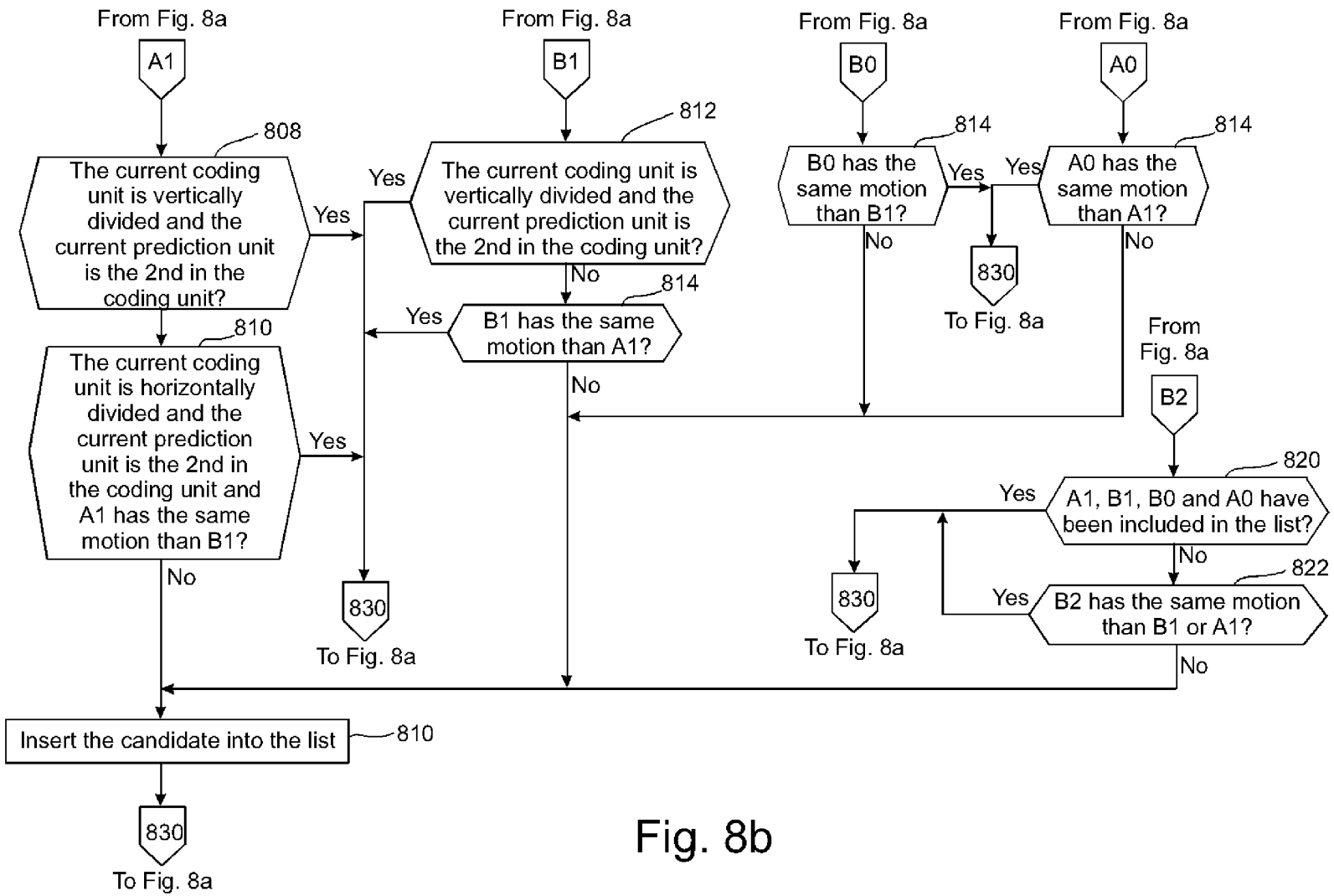


Fig. 8b

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000072

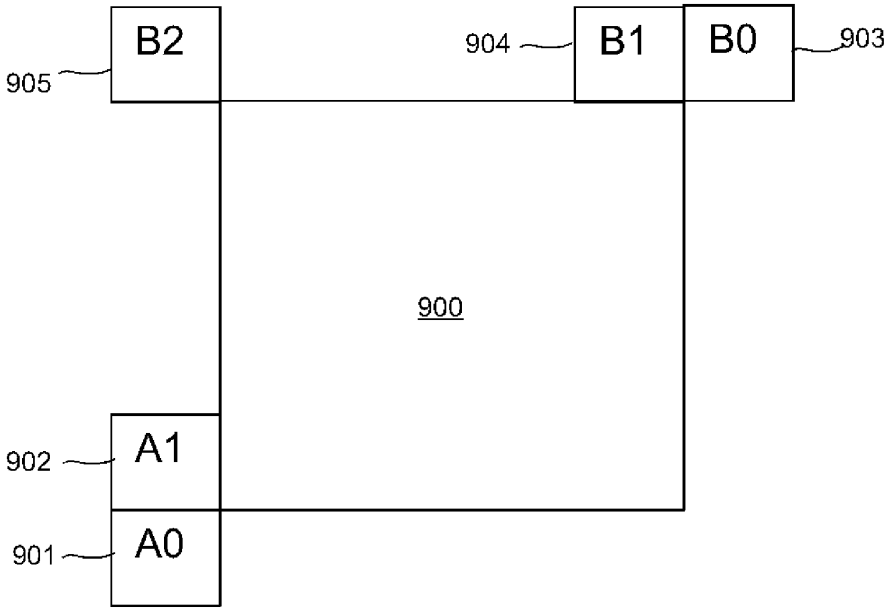


Fig. 9

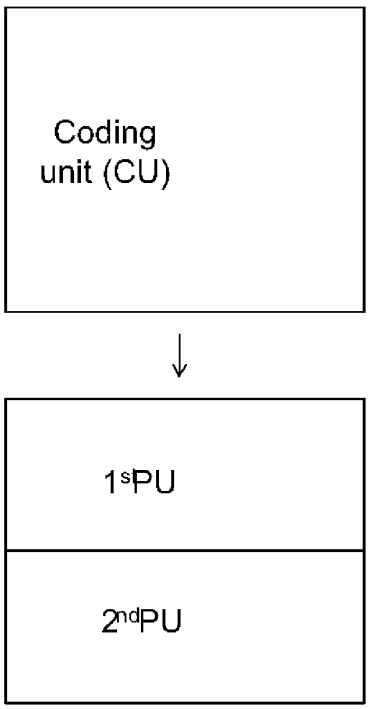


Fig. 10a

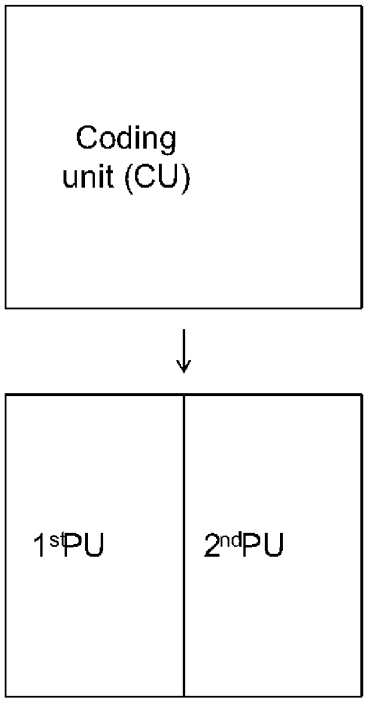


Fig. 10b

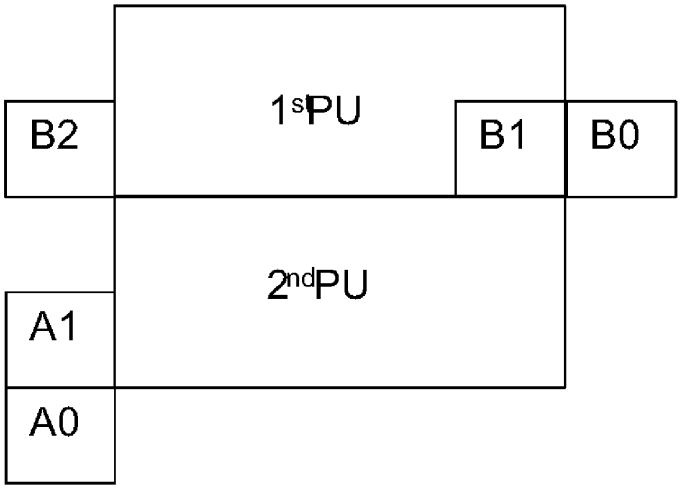


Fig. 11a

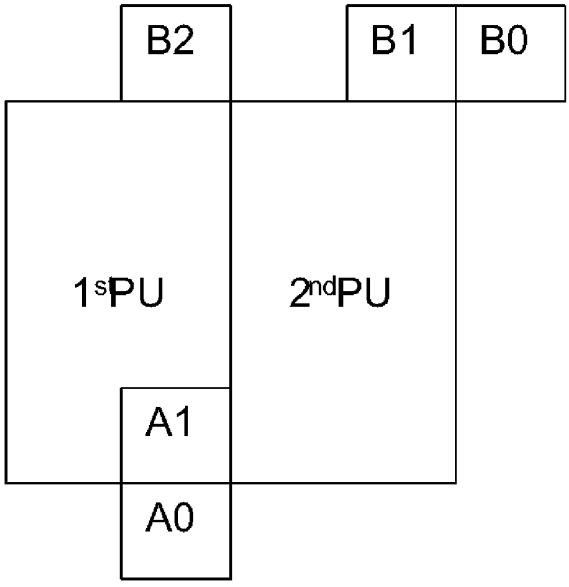


Fig. 11b

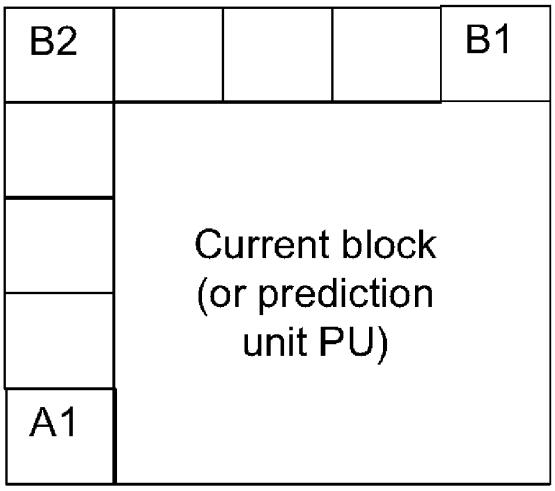


Fig. 12

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	METHOD FOR CODING AND AN APPARATUS
First Named Inventor/Applicant Name:	Mehmet Oguz BICI
Filer:	Thomas Joseph Arria/thao pham
Attorney Docket Number:	NC77198US-NP

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Utility application filing	1011	1	390	390
Utility Search Fee	1111	1	620	620
Utility Examination Fee	1311	1	250	250
Pages:				
Claims:				
Independent claims in excess of 3	1201	5	250	1250
Miscellaneous-Filing:				
Petition:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		2510

Electronic Acknowledgement Receipt

EFS ID:	14129936
Application Number:	13666680
International Application Number:	
Confirmation Number:	4782
Title of Invention:	METHOD FOR CODING AND AN APPARATUS
First Named Inventor/Applicant Name:	Mehmet Oguz BICI
Customer Number:	73658
Filer:	Thomas Joseph Arria/thao pham
Filer Authorized By:	Thomas Joseph Arria
Attorney Docket Number:	NC77198US-NP
Receipt Date:	01-NOV-2012
Filing Date:	
Time Stamp:	17:52:47
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$2510
RAM confirmation Number	4958
Deposit Account	500270
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 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	77198_ADS.pdf	837045 cc4bfb4c800fa962613531be16b54866bd271ba7	no	5
Warnings:					
Information:					
2		77198US-NP_Specification.pdf	231397 6e610cea7fea41e52a4bdd06cb679afbceee123e	yes	56

Multipart Description/PDF files in .zip description

Document Description	Start	End
Specification	1	45
Claims	46	55
Abstract	56	56

Warnings:					
Information:					
3	Drawings-only black and white line drawings	77198_drawings.pdf	157966 d0f31e08b0ec09fdccf729907fae7a17633a6799	no	13
Warnings:					
Information:					
4	Fee Worksheet (SB06)	fee-info.pdf	36587 5c191f1951b5032e2ac44f62defeeebc36a4ff21	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	1262995		

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.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 13/666,680

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	390
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	620
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	250
TOTAL CLAIMS (37 CFR 1.16(i))	20 minus 20 =	*			OR	x 62 =	0.00
INDEPENDENT CLAIMS (37 CFR 1.16(h))	8 minus 3 =	* 5				x 250 =	1250
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 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).						0.00
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0.00
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	2510

APPLICATION AS AMENDED - PART II

		(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($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	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	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
AMENDMENT B	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 found in the appropriate box in column 1.

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
13/666,680	11/01/2012	Mehmet Oguz BICI	NC77198US-NP

CONFIRMATION NO. 4782

73658
Nokia, Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086

FORMALITIES LETTER

OC000000057716268

Date Mailed: 11/26/2012

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing.

Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- A surcharge (for late submission of the basic filing fee, search fee, examination fee or inventor's oath or declaration) as set forth in 37 CFR 1.16(f) of $ **130** for a non-small entity, must be submitted.

SUMMARY OF FEES DUE:

Total fee(s) required within **TWO MONTHS** from the date of this Notice is $ **130** for a non-small entity
- $ **130** Surcharge.

Items Required To Avoid Processing Delays:

Applicant is notified that the above-identified application contains the deficiencies noted below. No period for reply is set forth in this notice for correction of these deficiencies. However, if a deficiency relates to the inventor's oath or declaration, the applicant must file an oath or declaration in compliance with 37 CFR 1.63, or a substitute statement in compliance with 37 CFR 1.64, executed by or with respect to each actual inventor no later than the expiration of the time period set in the "Notice of Allowability" to avoid abandonment. See 37 CFR 1.53(f).

- A properly executed inventor's oath or declaration has not been received for the following inventor(s):
  all
  Applicant may submit the inventor's oath or declaration at any time before the Notice of Allowance and Fee(s) Due, PTOL-85, is mailed.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000082

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web.
https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html

For more information about EFS-Web please call the USPTO Electronic Business Center at **1-866-217-9197** or visit our website at http://www.uspto.gov/ebc.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

/smunpanthovong/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000083

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	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
13/666,680	11/01/2012	2631	2510	NC77198US-NP	20	8

CONFIRMATION NO. 4782

73658
Nokia, Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086

FILING RECEIPT

OC000000057716267

Date Mailed: 11/26/2012

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Mehmet Oguz BICI, Tampere, TURKEY;
Jani LAINEMA, Tampere, FINLAND;
Kemal UGUR, Tampere, FINLAND;

Applicant(s)

Mehmet Oguz BICI, Tampere, TURKEY;
Jani LAINEMA, Tampere, FINLAND;
Kemal UGUR, Tampere, FINLAND;

Assignment For Published Patent Application

NOKIA CORPORATION, Espoo, FINLAND

Power of Attorney: None

Domestic Priority data as claimed by applicant

This appln claims benefit of 61/555,703 11/04/2011

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

If Required, Foreign Filing License Granted: 11/20/2012
The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 13/666,680**
Projected Publication Date: 05/09/2013
Non-Publication Request: No

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000084

Early Publication Request: No
Title

METHOD FOR CODING AND AN APPARATUS

Preliminary Class

375

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000085

the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage, facilitate, and accelerate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

To: IPR.Inhouse@nokia.com,brian.odonoghue@nokia.com,thao.pham@nokia.com
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 73658

Nov 26, 2012 05:21:34 AM

Dear PAIR Customer:

Nokia, Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 73658 , 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.
13666680	NTC.MISS.PRT	11/26/2012	NC77198US-NP
	APP.FILE.REC	11/26/2012	NC77198US-NP

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

PTO/SB/81 (01-06)
Approved for use through 12/31/2008. OMB 0651-0035
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.

POWER OF ATTORNEY and CORRESPONDENCE ADDRESS INDICATION FORM

Application Number	13/666680
Filing Date	November 1 2012
First Named Inventor	Mehmet Oguz BICI
Title	METHOD FOR CODING AND AN
Art Unit	TBD
Examiner Name	TBD
Attorney Docket Number	NC77198US-NP

I hereby revoke all previous powers of attorney given in the above-identified application.

I hereby appoint:

☑ Practitioners associated with the Customer Number: 73658

OR

☐ Practitioner(s) named below:

Name	Registration Number

as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Please recognize or change the correspondence address for the above-identified application to:

☑ The address associated with the above-mentioned Customer Number:

OR

☐ The address associated with Customer Number:

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 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

SIGNATURE of Applicant or Assignee of Record

Signature	/Thomas J. Arria/	Date	January 2, 2013
Name	Thomas J. Arria	Telephone	781-219-8760
Title and Company	Senior IPR Specialist, Nokia Corporation		

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 one forms are submitted.

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. 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 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.

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
15 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, 2013.

NOKIA CORPORATION

By:
Name: Harri Honkasalo
Title: Director,
Legal and Intellectual Property

By:
Name: Paul Melin
Title: Vice President,
Intellectual Property

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000089

PTO/SB/96 (01-08)
Approved for use through 03/31/2008. 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.

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: Mehmet Oguz BICI

Application No./Patent No.: 13/666680 Filed/Issue Date: November 1 2012

Entitled: METHOD FOR CODING AND AN APPARATUS

Nokia Corporation, a Corporation

(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that it is:

1. ☑ the assignee of the entire right, title, and interest; or

2. ☐ an assignee of less than the entire right, title and interest
(The extent (by percentage) of its ownership interest is_____ %)

in the patent application/patent identified above by virtue of either:

A. ☑ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

OR

B. ☐ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

2. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

3. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet.

☑ As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (*i.e.,* a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

/Thomas J. Arria/	January 2, 2013
Signature	Date
Thomas J. Arria	781-219-8760
Printed or Typed Name	Telephone Number
Senior IPR Specialist, Nokia Corporation	
Title	

This collection of information is required by 37 CFR 3.73(b). 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.

Docket No.: NC77198US-NP

ASSIGNMENT

Title of Invention	METHOD FOR CODING AND AN APPARATUS

This Assignment is directed to (check all that apply):

☒ U.S. Provisional No. 61/555703 filed on November 4, 2011

☒ U.S. Application No. 13/666680 filed on November 1, 2012

☐ which is a 371 National Stage of _____ filed on _____

As the below named inventor(s), I/we have invented certain improvements in the patent application for which I/we have filed an application for Letters Patent of the United States of America as identified above;

WHEREAS, I/we authorize the attorney of record to update this document to include Patent Office information as deemed necessary (i.e., filing date, serial number, etc.);

WHEREAS, NOKIA CORPORATION, a corporation organized under the laws of Finland, having its principal office in Espoo, Finland (hereinafter referred to as "ASSIGNEE"), is desirous of acquiring the entire right, title and interest in and under the said invention and the said application, and in and to any and all Letters Patent which shall be granted therefore in the United States of America and in any and all foreign countries;

NOW, THEREFORE, for good and valuable consideration, the receipt and sufficiency of which is hereby acknowledged, I/we have sold and do hereby sell, assign, transfer and convey unto said ASSIGNEE, its successors, assigns and legal representatives, the entire right, title and interest in and to said invention and application, and in all divisionals, reissues, substitutions, continuations, continuation-in-part and, in any and all Letters Patents of the United States of America and all foreign countries or reissues, reexaminations, or extensions thereof which may be granted therefore or thereon, for the full end of the term for which said Letters Patent may be granted, together with the right to claim the priority of said application in all foreign countries in accordance with the International Convention, the same to be held and enjoyed by said ASSIGNEE, its successors and assigns, as fully and entirely as the same would have been held and enjoyed by me if this assignment and sale had not been made.

I/WE ALSO HEREBY authorize and request the Commissioner of Patents and Trademarks to issue all patents for said invention, or patents resulting therefrom to the said ASSIGNEE of my/our entire right, title and interest.

I/WE FURTHER HEREBY sell and assign to said ASSIGNEE, its successors, assigns and legal representatives the full and exclusive rights, title and interest to the invention disclosed in said application throughout the world, including, without limitation, the right to file applications and obtain patents, utility models, industrial models and designs for said invention in its own name throughout the world including all rights of priority, all rights to publish cautionary notices reserving ownership of said invention and all rights to register said invention in appropriate registries. I/we further agree to execute any and all powers of attorney, applications, assignments, declarations, affidavits, and any other papers in connection therewith necessary to perfect such rights, title and interest in ASSIGNEE, its successors, assigns and legal representatives.

I/WE HEREBY covenant that I/we have not and I/we will not execute any agreement in conflict herewith.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000091

Docket No.: NC77198US-NP

ASSIGNMENT

I/WE HEREBY further covenant, and agree to bind my/our heirs, legal representatives, and assigns, promptly to communicate to said ASSIGNEE or its representatives any facts known to me relating to said invention, to testify in any interference or legal proceedings involving said invention, to execute any additional papers which may be requested to confirm the right of the assignee, its representatives, successors, or assigns to secure patent or similar protection for the said invention in all countries and to vest in the assignee complete title to the said invention and Letters Patent, without further compensation, but at the expense of said ASSIGNEE, its successors, assigns, and other legal representatives.

IN WITNESS WHEREOF, I/we have hereunto set my/our hand and seal on the date indicated below.

Inventor: Mehmet Oguz BICI Date: NOV -05 - 2012

Signature: _____ Residence: Tampere, Finland

Witness Signature: _____

Printed Name: PAYMAN AFLAKI Date: 5, 11, 2012

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000092

ASSIGNMENT

IN WITNESS WHEREOF, I/we have hereunto set my/our hand and seal on the date indicated below.

Inventor: Jani LAINEMA Date: NOV-02-2012

Signature: Residence: Tampere, Finland

Witness Signature:

Printed Name: PAYMAN AFLAKI Date: 2/11/2012

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000093

ASSIGNMENT

IN WITNESS WHEREOF, I/we have hereunto set my/our hand and seal on the date indicated below.

Inventor: Kemal UGUR Date: Nov-02-2012

Signature: Residence: Tampere, Finland

Witness Signature:

Printed Name: PAYMAN AFLAKI Date: 2/11/2012

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000094

Electronic Patent Application Fee Transmittal

Application Number:	13666680
Filing Date:	01-Nov-2012
Title of Invention:	METHOD FOR CODING AND AN APPARATUS
First Named Inventor/Applicant Name:	Mehmet Oguz BICI
Filer:	Thomas Joseph Arria/thao pham
Attorney Docket Number:	NC77198US-NP

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:				
Late filing fee for oath or declaration	1051	1	130	130
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**130**

Electronic Acknowledgement Receipt

EFS ID:	14597236
Application Number:	13666680
International Application Number:	
Confirmation Number:	4782
Title of Invention:	METHOD FOR CODING AND AN APPARATUS
First Named Inventor/Applicant Name:	Mehmet Oguz BICI
Customer Number:	73658
Filer:	Thomas Joseph Arria/thao pham
Filer Authorized By:	Thomas Joseph Arria
Attorney Docket Number:	NC77198US-NP
Receipt Date:	02-JAN-2013
Filing Date:	01-NOV-2012
Time Stamp:	15:16:04
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	1609
Deposit Account	500270
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	Oath or Declaration filed	77198_executed_declaration.pdf	268093 0df10ceb9f01d2e1b415cfb34f4031e3cf6351c8	no	3
Warnings:					
Information:					
2		77198_POA_373b.pdf	378033 2a71d054ee4ec5f0e34ccb017d1ff096fe0b6141	yes	7
	Multipart Description/PDF files in .zip description				
	Document Description		**Start**	**End**	
	Power of Attorney		1	2	
	Assignee showing of ownership per 37 CFR 3.73.		3	7	
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	30297 4e38822359718c4cd9a4820faa6e6fd59c1598db	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	676423		

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.

PTO/AIA/01 (06-12)
Approved for use through 01/31/2014. 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.

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHOD FOR CODING AND AN APPARATUS

As the below named inventor, I hereby declare that:

This declaration is directed to:

☐ The attached application, or

☒ United States application or PCT international application number 13/666680

filed on November 1, 2012.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Mehmet Oguz BICI Date (Optional): NOV - 05 - 2012

Signature: _____

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form. Use an additional PTO/AIA/01 form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. 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 1 minute 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.

PTO/AIA/01 (06-12)
Approved for use through 01/31/2014. 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.

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHOD FOR CODING AND AN APPARATUS

As the below named inventor, I hereby declare that:

This declaration is directed to:

☐ The attached application, or

☒ United States application or PCT international application number 13/666680

filed on November 1, 2012.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than **five** (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Jani LAINEMA Date (Optional): NOV-02-2012

Signature: Jani

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form. Use an additional PTO/AIA/01 form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. 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 1 minute 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.

PTO/AIA/01 (06-12)
Approved for use through 01/31/2014. 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.

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHOD FOR CODING AND AN APPARATUS

As the below named inventor, I hereby declare that:

This declaration is directed to:

☐ The attached application, or

☒ United States application or PCT international application number 13/666680

filed on November 1, 2012.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Kemal UGUR Date (Optional): Nov-02-2012

Signature: 

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form. Use an additional PTO/AIA/01 form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. 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 1 minute 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.

PTO/SB/08a (01-10)
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 contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**		
	Application Number	13666680
	Filing Date	2012-11-01
	First Named Inventor	Mehmet Oguz BICI
	Art Unit	2631
	Examiner Name	TBD
	Attorney Docket Number	NC77198US-NP

U.S.PATENTS						Remove
Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					
If you wish to add additional U.S. Patent citation information please click the Add button.						Add

U.S.PATENT APPLICATION PUBLICATIONS						Remove
Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20120307905	A1	2012-12-06	Sunyeon et al.	
	2	20110170602	A1	2011-07-14	Tammy et al.	
If you wish to add additional U.S. Published Application citation information please click the Add button.						Add

FOREIGN PATENT DOCUMENTS								Remove
Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1	2011062392	WO	A2	2011-05-26	SK Telecom Co. Ltd. et al.		☐
If you wish to add additional Foreign Patent Document citation information please click the Add button								Add

NON-PATENT LITERATURE DOCUMENTS	Remove

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000102

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	13666680
	Filing Date	2012-11-01
	First Named Inventor	Mehmet Oguz BICI
	Art Unit	2631
	Examiner Name	TBD
	Attorney Docket Number	NC77198US-NP

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	International Search Report and Written Opinion received for corresponding Patent Cooperation Treaty Application No. PCT/FI2012/051070, dated March 27, 2013, 12 pages.	☐
	2	Oudin, S. et al.; "Block Merging for Quadtree-Based Video Coding", IEEE Int. Conf. on Multimedia and Expo, July 11-15, 2011, 6 pages.	☐
	3	Sullivan, G. J.; "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 22, No. 12, December 2012, pp. 1649-1668.	☐

If you wish to add additional non-patent literature document citation information please click the Add button **Add**

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	13666680
	Filing Date	2012-11-01
	First Named Inventor	Mehmet Oguz BICI
	Art Unit	2631
	Examiner Name	TBD
	Attorney Docket Number	NC77198US-NP

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☐ Fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ None

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Thomas J. Arria/	Date (YYYY-MM-DD)	2013-04-22
Name/Print	Thomas J. Arria	Registration Number	60223

This collection of information is required by 37 CFR 1.97 and 1.98. 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 1 hour 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.**

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 the Freedom of Information Act requires disclosure of these record s.

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 inspections 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.

PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

To:
Nokia Corporation
IPR Department Jussi Jaatinen
Keilalahdentie 4
FI- 02150 Espoo
FINLAND

PCT

NOTIFICATION OF TRANSMITTAL OF THE INTERNATIONAL SEARCH REPORT AND THE WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY, OR THE DECLARATION

(PCT Rule 44.1)

	Date of mailing *(day/month/year)* 27 March 2013 (27.03.2013)
Applicant's or agent's file reference NC77198WO	**FOR FURTHER ACTION** See paragraphs 1 and 4 below
International application No. PCT/FI2012/051070	International filing date *(day/month/year)* 02 November 2012 (02.11.2012)
Applicant NOKIA CORPORATION	

1. ☒ The applicant is hereby notified that the international search report and the written opinion of the International Searching Authority have been established and are transmitted herewith.

   Filing of amendments and statement under Article 19:
   The applicant is entitled, if he so wishes, to amend the claims of the international application (see Rule 46):

   When? The time limit for filing such amendments is normally two months from the date of transmittal of the international search report.

   Where? Directly to the International Bureau of WIPO, 34 chemin des Colombettes 1211 Geneva 20, Switzerland, Facsimile No.: +41 22 338 82 70

   For more detailed instructions, see *PCT Applicant's Guide*, International Phase, paragraphs 9.004 – 9.011.

2. ☐ The applicant is hereby notified that no international search report will be established and that the declaration under Article 17(2)(a) to that effect and the written opinion of the International Searching Authority are transmitted herewith.

3. ☐ **With regard to any protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:

   ☐ the protest together with the decision thereon has been transmitted to the International Bureau together with the applicant's request to forward the texts of both the protest and the decision thereon to the designated Offices.

   ☐ no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Reminders**

   The applicant may submit comments on an informal basis on the written opinion of the International Searching Authority to the International Bureau. The International Bureau will send a copy of such comments to all designated Offices unless an international preliminary examination report has been or is to be established. Following the expiration of 30 months from the priority date, these comments will also be made available to the public.

   Shortly after the expiration of **18 months** from the priority date, the international application will be published by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau before the completion of the technical preparations for international publication (Rules 90*bis*.1 and 90*bis*.3).

   Within **19 months** from the priority date, but only in respect of some designated Offices, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase **until 30 months** from the priority date (in some Offices even later); otherwise, the applicant must, **within 20 months** from the priority date, perform the prescribed acts for entry into the national phase before those designated Offices.

   In respect of other designated Offices, the time limit of **30 months** (or later) will apply even if no demand is filed within 19 months.

   For details about the applicable time limits, Office by Office, see www.wipo.int/pct/en/texts/time_limits.html and the *PCT Applicant's Guide*, National Chapters.

Name and mailing address of the ISA/FI National Board of Patents and Registration of Finland P.O. Box 1160, FI-00101 HELSINKI, Finland Facsimile No. +358 9 6939 5328	Authorized officer Timo Laakso Telephone No. +358 9 6939 500

Form PCT/ISA/220 (July 2010)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000106

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference NC77198WO	**FOR FURTHER ACTION**	see Form PCT/ISA/220 as well as, where applicable, item 5 below.
International application No. PCT/FI2012/051070	International filing date *(day/month/year)* 02 November 2012 (02.11.2012)	(Earliest) Priority Date *(day/month/year)* 04 November 2011 (04.11.2011)
Applicant NOKIA CORPORATION		

This international search report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This international search report consists of a total of 4 sheets.

☒ It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**
   a. With regard to the **language**, the international search was carried out on the basis of:
      ☒ the international application in the language in which it was filed.
      ☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).
   b. ☐ This international search report has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43.6*bis*(a)).
   c. ☐ With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, see Box No. I.
2. ☐ **Certain claims were found unsearchable** (see Box No. II).
3. ☐ **Unity of invention is lacking** (see Box No. III).
4. With regard to the **title**,
   ☐ the text is approved as submitted by the applicant.
   ☒ the text has been established by this Authority to read as follows:
   METHOD FOR VIDEO CODING AND APPARATUS
5. With regard to the **abstract**,
   ☒ the text is approved as submitted by the applicant.
   ☐ the text has been established, according to Rule 38.2, by this Authority as it appears in Box No. IV. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.
6. With regard to the **drawings**,
   a. the figure of the **drawings** to be published with the abstract is Figure No. 6b
      ☐ as suggested by the applicant.
      ☒ as selected by this Authority, because the applicant failed to suggest a figure.
      ☐ as selected by this Authority, because this figure better characterizes the invention.
   b. ☐ none of the figures is to be published with the abstract.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000107

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2012/051070

A. CLASSIFICATION OF SUBJECT MATTER

See extra sheet

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: G06T, H04N

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

FI, SE, NO, DK

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI, Google Scholar, IEEE Xplore

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2011062392 A2 (SK TELECOM CO LTD et al.) 26 May 2011 (26.05.2011) abstract, claim 1 & US 2012307905 A1 (KIM SUNYEON et al) 6 December 2012 (06.12.2012)	1-20
A	US 2011170602 A1 (LEE TAMMY et al.) 14 July 2011 (14.07.2011) the whole document	1-20
A	OUDIN, S. et al.: "Block merging for quadtree-based video coding", IEEE Int. Conf. on Multimedia and Expo, 11-15 July 2011, 6p the whole document	1-20
T	SULLIVAN, G. J.: "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Trans. on Circuits and Systems for Video technology, vol. 22, no. 12, Dec. 2012, pp. 1649-1668 the whole document	1-20

☐ 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
26 March 2013 (26.03.2013)	27 March 2013 (27.03.2013)

Name and mailing address of the ISA/FI	Authorized officer
National Board of Patents and Registration of Finland P.O. Box 1160, FI-00101 HELSINKI, Finland	Timo Laakso
Facsimile No. +358 9 6939 5328	Telephone No. +358 9 6939 500

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/FI2012/051070

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
WO 2011062392 A2	26/05/2011	US 2012307905 A1	06/12/2012
		CN 102714720 A	03/10/2012
		KR 20110054592 A	25/05/2011
US 2011170602 A1	14/07/2011	US 2013044815 A1	21/02/2013
		MX 2012008229 A	17/08/2012
		EP 2524507 A2	21/11/2012
		CN 102792697 A	21/11/2012
		US 2012155542 A1	21/06/2012
		SG 182491 A1	30/08/2012
		WO 2011087321 A2	21/07/2011
		CA 2787006 A1	21/07/2011
		AU 2011205896 A1	30/08/2012
		KR 20110083365 A	20/07/2011

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000109

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2012/051070

CLASSIFICATION OF SUBJECT MATTER

Int.Cl.
H04N 7/26 (2006.01)
H04N 7/50 (2006.01)
H04N 7/34 (2006.01)
G06T 9/00 (2006.01)

Form PCT/ISA/210 (extra sheet)

PATENT COOPERATION TREATY

From the
INTERNATIONAL SEARCHING AUTHORITY

To:
Nokia Corporation
IPR Department Jussi Jaatinen
Keilalahdentie 4
FI- 02150 Espoo
FINLAND

PCT

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

(PCT Rule 43*bis*.1)

	Date of mailing *(day/month/year)*: 27 March 2013 (27.03.2013)
Applicant's or agent's file reference: NC77198WO	**FOR FURTHER ACTION**: See paragraph 2 below

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/FI2012/051070	02 November 2012 (02.11.2012)	04 November 2011 (04.11.2011)

International Patent Classification (IPC) or both national classification and IPC: See supplemental box

Applicant: NOKIA CORPORATION

1. This opinion contains indications relating to the following items:

- [x] Box No. I Basis of the opinion
- [ ] 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
- [x] Box No. V Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step and industrial applicability; citations and explanations supporting such statement
- [ ] Box No. VI Certain documents cited
- [x] Box No. VII Certain defects in the international application
- [ ] Box No. VIII Certain observations on the international application

2. **FURTHER ACTION**

If a demand for international preliminary examination is made, this opinion will be considered to be a written opinion of the International Preliminary Examining Authority ("IPEA") except that this does not apply where the applicant chooses an Authority other than this one to be the IPEA and the chosen IPEA has notified the International Bureau under Rule 66.1*bis*(b) that written opinions of this International Searching Authority will not be so considered.

If this opinion is, as provided above, considered to be a written opinion of the IPEA, the applicant is invited to submit to the IPEA a written reply together, where appropriate, with amendments, before the expiration of 3 months from the date of mailing of Form PCT/ISA/220 or before the expiration of 22 months from the priority date, whichever expires later.

For further options, see Form PCT/ISA/220.

Name and mailing address of the ISA/FI	Date of completion of this opinion	Authorized officer
National Board of Patents and Registration of Finland P.O. Box 1160, FI-00101 HELSINKI, Finland	26 March 2013 (26.03.2013)	Timo Laakso
Facsimile No. +358 9 6939 5328		Telephone No. +358 9 6939 500

Form PCT/ISA/237 (cover sheet) (July 2011)

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/FI2012/051070

Box No. I Basis of this opinion

1. With regard to the **language**, this opinion has been established on the basis of:

   ☒ the international application in the language in which it was filed.

   ☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

2. ☐ This opinion has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43*bis*.1(a))

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, this opinion has been established on the basis of a sequence listing filed or furnished:

   a. (means)

      ☐ on paper
      ☐ in electronic form

   b. (time)

      ☐ in the international application as filed
      ☐ together with the international application in electronic form
      ☐ subsequently to this Authority for the purposes of search

4. ☐ In addition, in the case that more than one version or copy of a sequence listing has been filed or furnished, the required statements that the information in the subsequent or additional copies is identical to that in the application as filed or does not go beyond the application as filed, as appropriate, were furnished.

5. Additional comments:

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000112

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/FI2012/051070

Box No. V **Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement**

1. Statement

Novelty (N)	Claims	1-20	YES
	Claims		NO
Inventive step (IS)	Claims	1-20	YES
	Claims		NO
Industrial applicability (IA)	Claims	1-20	YES
	Claims		NO

2. Citations and explanations:

2.1 Documents cited in the International Search Report

D1: WO 2011062392 A2 (SK TELECOM CO LTD et al.) 26 May 2011 (26.05.2011)
D2: US 2011170602 A1 (LEE TAMMY et al.) 14 July 2011 (14.07.2011)
D3: OUDIN, S. et al.: "Block merging for quadtree-based video coding", IEEE Int. Conf. on Multimedia and Expo, 11-15 July 2011, 6p.
D4: SULLIVAN, G. J.: "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Trans. on Circuits and Systems for Video technology, vol. 22, no. 12, Dec. 2012, pp. 1649-1668.

It is assumed that the contents of WO2011062392 correspond to those of US2012307905 which belongs to the same patent family.

2.2 Claimed invention

As described in claim 1, the invention relates to a method for video coding, comprising:

(F1) receiving a block of pixels including a prediction unit;
(F2) determining a set of spatial motion vector prediction candidates for the block of pixels; the spatial motion vector prediction candidates being provided with motion information;
(F3) selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;
(F4) determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;
(F5) comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;
(F6) if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

(F1)...(F6) denote the technical features of the invention. Independent claims 9 and 15-20 define another method, two apparatuses, two storage mediums and two apparatuses with the corresponding technical features.

In addition, dependent claims 2-8 and 10-14 specify preferred embodiments of the invention.

The objective of the invention is to provide an improved arrangement for video coding for mobile communications systems.

Continued to next page

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000113

Supplemental Box

Continuation of: Box V (1 / 2)

2.3 Novelty under PCT Article 33(2)

D1 (abstract, claim 1) discloses a method of encoding a motion vector that includes: selecting one of a plurality of predicted candidate motion vector sets by using motion information of neighboring blocks of a current block; selecting one of predicted candidate motion vectors within a selected predicted candidate motion vector set, as a predicted motion vector; encoding a differential motion vector representing a difference between a current motion vector or motion vector of the current block and a selected predicted motion vector; and encoding a predicted motion vector index indicating the selected predicted motion vector. As a motion vector is encoded after selecting an efficient predicted candidate motion vector set, the size of a differential vector to be encoded can be reduced without necessarily encoding additional information to indicate which set of predicted candidate motion vectors has been selected, resulting in improved compression efficiency of motion vectors and in turn the improved video compression efficiency.

D2 introduces methods and apparatuses for encoding and decoding a motion vector. The method of encoding a motion vector includes: selecting a mode from among a first mode in which information indicating a motion vector predictor of at least one motion vector predictor is encoded and a second mode in which information indicating generation of a motion vector predictor based on pixels included in a previously encoded area adjacent to a current block is encoded; determining a motion vector predictor of the current block according to the selected mode and encoding information about the motion vector predictor of the current block; and encoding a difference vector between a motion vector of the current block and the motion vector predictor of the current block.

D3 presents an approach for block merging that removes redundancies by using a single parameter for a whole motion-compensated region of contiguous blocks. The proposed technique provides improvement over the specified H.264/AVC video coding standard and is included in the novel High Efficiency Video Coding (HEVC) standard.

D4 is a tutorial overview article of the novel HEVC video coding standard.

Documents D1-D3 represent the most relevant prior art. D4 is a later document not in conflict with the application that is cited to understand the theory underlying the invention.

2.3.1 Independent claims

Document D1 is regarded as the prior art closest to the subject matter of independent claim 1.

Document D1 (abstract, claim 1) discloses a methods for encoding a motion vector by selecting a set of predicted candidate motion vectors by using motion information of neighboring blocks of a current block. The subject matter of claim 1 differs from this in that merging of spatial motion vectors is not considered in D1 as in feature (F6) of claim 1. The subject matter of claim 1 is therefore novel. The same applies to independent claims 9 and 15-20 with the corresponding technical features.

2.3.2 Dependent claims

Because independent claims 1 and 9 are novel, also dependent claims 2-8 and 10-14 are novel.

Continued to next page

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000114

Supplemental Box

Continuation of: Box V (2 / 2)

2.4 Inventive step under PCT Article 33(3)

2.4.1 Independent claims

The solution presented in claim 1 is neither described in nor made obvious by the available prior art. Thus, claim 1 involves an inventive step. The same applies to independent claims 9 and 15-20 with the corresponding technical features.

2.4.2 Dependent claims

Because independent claims 1 and 9 involve an inventive step, also dependent claims 2-8 and 10-14 involve an inventive step.

2.5 Industrial applicability under PCT Article 33(4)

Claims 1-20 meet the requirement of industrial applicability because the claimed subject matter can be made or used in industry.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000115

Box No. VII Certain defects in the international application

The following defects in the form or contents of the international application have been noted:

Contrary to the requirements of PCT Rule 5.1(a)(ii), no relevant prior art is indicated in the description, nor are the documents reflecting such art identified therein.

Claims are not drafted in a two-part form in accordance with PCT Rule 6.3(b), which in the present case would be appropriate, with those features known from the prior art placed in the preamble (PCT Rule 6.3(b)(i)) and the remaining features included in the characterising part (PCT Rule 6.3(b)(ii)).

The features of the claims are not provided with reference signs placed in parentheses (PCT Rule 6.2(b)).

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000116

Supplemental Box

In case **the space in any of the preceding boxes is not sufficient.**

Continuation of: International Patent Classification (IPC)

Int.Cl.
H04N 7/26 (2006.01)
H04N 7/50 (2006.01)
H04N 7/34 (2006.01)
G06T 9/00 (2006.01)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000117

BLOCK MERGING FOR QUADTREE-BASED VIDEO CODING

S. Oudin[1], *P. Helle*[1], *J. Stegemann*[1], *C. Bartnik*[1], *B. Bross*[1], *D. Marpe*[1], *H. Schwarz*[1], *and T. Wiegand*[1,2]

[1]Image Processing Department
Fraunhofer Institute for Telecommunications - Heinrich Hertz Institute
Einsteinufer 37, 10587 Berlin, Germany

[2]Image Communication Chair
Department of Telecommunication Systems
Technical University of Berlin
Einsteinufer 17, 10587 Berlin, Germany

ABSTRACT

Quadtree-based block partitioning together with motion-compensated prediction has proven to be an efficient approach in video compression. However, when dealing with spatially neighboring blocks in uniformly displaced regions, quadtree-based partitioning may lead to redundant sets of transmitted motion parameters. This paper proposes and describes a simple but efficient block merging algorithm that aims at removing those redundancies by using only a single parameter set for a whole motion-compensated region of contiguous blocks. Simulation results show that our proposed merging technique works more efficiently than the conceptually similar direct mode as, e.g., specified in H.264/AVC. Due its efficiency and simplicity, our proposed merging approach has been adopted into the first test model of the high efficiency video coding (HEVC) standardization project, as currently pursued by ITU-T VCEG and ISO/IEC MPEG.

Index Terms— motion compensation, quadtree partition, block merging, H.265, HEVC

1. INTRODUCTION

In image and video coding, pictures and their corresponding sets of sample arrays are usually decomposed into blocks such that each block is associated with a particular set of coding parameters. In H.264/AVC [1], e.g., each block is either spatially or temporally predicted and the resulting prediction residual is represented by using transform coding. Temporal or inter-picture prediction typically uses samples of one or two displaced blocks in previously decoded and reconstructed reference pictures for the samples of a given picture block. The motion vectors corresponding to these displacements are typically coded in a differential way by using an appropriately specified motion-vector predictor. Thus, without any further provision, one or two sets of motion parameters have to be signaled for each inter-predicted block as motion information with each set consisting of a motion-vector difference and an index specifying the corresponding reference picture.

In H.264/AVC, macroblocks or sub-macroblocks of bi-predictively coded areas can also be coded in so-called *direct mode* without the need of transmitting any additional motion information [1, 2]. The direct mode infers the motion information for prediction from previously decoded blocks. In addition, if no prediction residual signal is transmitted for a direct-coded macroblock, it is also referred to as *skipped*, and skipped macroblocks can be indicated very efficiently as for the *skip mode* in H.264/AVC.

In the present paper, we introduce a coding tool called *block merging* which is conceptually similar to the direct mode of H.264/AVC. Block merging has been designed as part of a novel quadtree-based video coding approach [3, 4]. It exploits the spatial redundancy of motion information in neighboring blocks belonging to potentially different branches and levels in the quadtree hierarchy and it operates in a way that is conceptually similar to what has already been introduced in previous related publications [5, 6, 7].

The presented block merging algorithm has been first introduced as a feature of the Fraunhofer HHI response [3] to the joint Call for Proposals (CfP) on high efficiency video coding (HEVC) technology that was issued by ITU-T VCEG and ISO/IEC MPEG [8]. The intention of the CfP was to identify video coding technology with substantially higher compression capabilities than the existing H.264/AVC standard.

The rest of the paper is organized as follows. The next section briefly introduces the HHI proposal for HEVC, also called *HHI video codec* in the sequel. Section 3 deals with quadtree based partitioning and its drawbacks. The proposed block merging algorithm is described in detail in Section 4. Conducted experiments and their results are presented in Sections 5 and 6. Section 7 concludes this paper.

2. OVERVIEW OF THE HHI VIDEO CODEC

The HHI video codec is based on the hybrid video coding approach using temporal and spatial prediction followed by transform coding of the residual and entropy coding [3, 4]. In this video codec, the size of the prediction blocks can be adaptively chosen by using a quadtree-based partitioning. The block size used for DCT-based coding of the residual signal is also derived by using quadtree-based partitioning of the corresponding prediction block. The concept of a macroblock

978-1-61284-350-6/11/$26.00 ©2011 IEEE

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000118

as the basic processing unit in standardized video coding [1] is generalized to what is called a *coding tree block* (CTB). Fig. 1(a) shows an example of a coding tree block for luma samples (solid bold outline) and how it is subdivided into prediction blocks (solid lines) and transforms blocks (dashed lines). In Fig. 1(b), the corresponding quadtree structure for CTB partitioning is shown. Each node of the so-called prediction quadtree, as shown in solid lines in the example of Fig. 1(b), which corresponds to a prediction block and its related residual signal, can be further split into transform blocks of smaller size than the given prediction block size. In the example of Fig. 1, this is illustrated by the dashed blocks and the corresponding dashed nested residual quadtrees.

Block merging, as further described below, enables the creation of regions composed of neighboring prediction blocks with identical motion information. This motion information is composed of one or two sets of a motion-vector difference and an index specifying the corresponding reference picture. However, the dimensions of the transform blocks for residual coding are still constrained by corresponding residual quadtree partition, independent of the merging operation. The direct mode which is used in our simulation as a reference is similar to the spatial one specified in H.264/AVC [1]. This spatial direct mode can be applied to inter-partitions in pictures using multi-hypothesis prediction. For these inter-partitions, a flag is signaled indicating whether the partition is coded in direct mode or not. Interleaved Motion Vector Predictor (IMVP), as introduced in [3, 4], interleaves the prediction and coding for the components of a motion vector. More precisely, it uses the vertical component in order to predict its horizontal component. In the HHI video codec, no explicit skip mode is used. However, for each leaf in the residual quadtree, there is a flag which signals explicitly the presence of non-vanishing residual coefficients in the corresponding block. For more details of the HHI video codec, the reader is referred to [3, 4].

3. BACKGROUND AND PROBLEM FORMULATION

The quadtree data structure provides a suitable framework in video compression for optimizing the tradeoff between model accuracy (e.g., given by distortion D) and model coding cost, typically measured in bit rate R. The existence of simple tree pruning algorithms optimizing a Lagrangian functional

$$D + \lambda * R,$$

allows the development of compression algorithms which are based on quadtrees and related representations. The HHI video codec is a quadtree-structured video coding approach with syntax elements carrying the subdivision information for blocks of different types. Such syntax elements may be related to subdivision of blocks for the purpose of prediction or transform coding, as already outlined in Section 2 above.

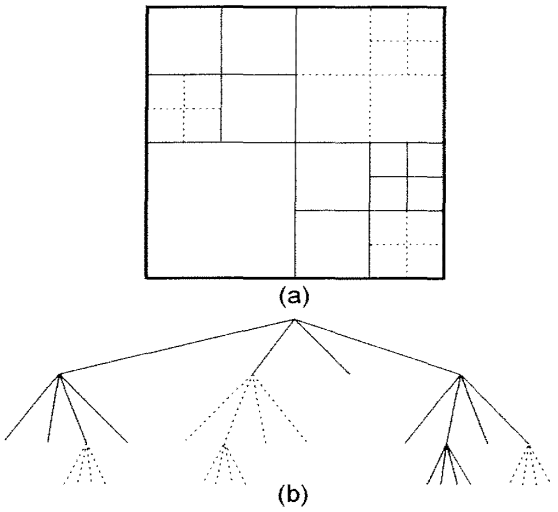


Fig. 1. (a) Example of coding tree block (solid bold) into prediction blocks (solid) and transform blocks (dashed) of variable size. (b) Corresponding nested quadtree structure.

However, the concept of quadtree-based image and video coding is intrinsically tied to certain drawbacks, as analyzed in [5]. For instance, the systematic decomposition of every block into four child blocks does not allow to jointly represent child blocks that belong to two different parent blocks. Also, if a given block is divided into four child blocks, all child blocks are typically coded separately, even if two or three of them may share the same coding parameters.

Thus, the sub-optimal rate-distortion properties of an initial quadtree-based subdivision can be substantially improved [5]. Block merging has been introduced in order to compensate the aforementioned drawbacks by enabling the merging of spatially neighboring blocks with identical coding parameters. As a consequence, block merging reduces the amount of redundant sets of transmitted coding parameters that are typically generated when using a quadtree-based block partitioning algorithm. In our proposed block merging algorithm, as further described below, specific syntax elements are introduced and coded together with the temporal prediction parameters for each leaf node of the prediction quadtree.

Merging is particularly well-suited for regions of homogeneous motion that can be sufficiently described by using a single motion parameter set. For example, a video signal often contains large spatially adjacent regions, each having homogeneous motion parameters that differ significantly among the regions. Inside each such region, the corresponding blocks can be merged, even if the maximum allowed block size for quadtree-based partitioning is reached.

Merging enables the creation of regions with arbitrary

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000119

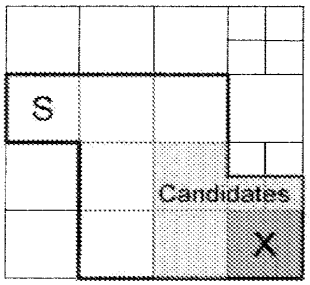


Fig. 2. A quadtree segmentation with multiple blocks merged into one region (bold line). Block S contains the prediction information for the region. Blocks in the gray shaded region are the candidates for block X to be merged with.

shapes. One illustrating example is shown in Fig. 2; other examples as outcome of our simulations are shown in Figs. 5–6. Ideally, the boundary of each region should coincide with the motion discontinuities in the given video signal. With quadtree-based block partitioning combined with merging, the picture can be subdivided into smaller and smaller blocks thereby approximating the motion boundary and minimizing the size of partitions containing the boundary. Subsequently, each created quadtree leaf block can be merged into a region on either side of the boundary.

4. THE PROPOSED BLOCK MERGING ALGORITHM

In order to illustrate the operation of our proposed block merging algorithm, Fig. 2 shows an example of a quadtree segmentation. The purpose of block merging, as already explained, is to compensate for the disadvantages of an initial quadtree-based subdivision by reducing the redundant sets of transmitted coding parameters. This is achieved by creating regions composed of neighboring prediction blocks with identical motion information. Each region must contain at least one block that is not merged, seeding the new motion information. In Fig. 2, block S (for seed) corresponds to such a particular block. Block X in Fig. 2 corresponds to the current block to be en-/decoded. All the blocks en-/decoded before the current block along the block scanning pattern form the set of causal blocks. Note that within each CTB, the subblocks are traversed in depth-first order. In any case, the block to be en-/decoded can be merged only with one of its causal neighboring blocks. In Fig. 2, these blocks are highlighted by the gray shaded region.

A list of candidate sets of motion information is created using the motion information collected from these neighboring blocks. If the block X is merged with one of these candidates, it means that block X copies and uses the same motion information of one of its neighbors. In order to identify this motion information in the list, an index is signaled for the block X. The particular design of the block merging algorithm used to obtain simulation results for this paper only uses the motion parameter sets of the left and top neighboring blocks as candidates. Thus, the candidate list contains at most two entries.

The new syntax elements merge_flag and merge_index are introduced for signalization. The binary valued merge_flag is signaled for each motion-compensated partition block prior to any prediction parameter. The list of merge candidates may be empty because, e.g., no neighboring block is predicted by using motion compensation. In this case, block X cannot be merged and merge_flag is not transmitted. Otherwise, if merge_flag is equal to 0, the current block X is not merged with any of its neighboring candidate blocks and motion parameters for X are signaled explicitly. If the motion information of two different merge candidates are identical, the list of candidate sets of motion information can be reduced. In this way, the list of candidates of motion information is only composed of motion information which are different and thus, its size is as small as possible. If there is only one candidate in the list, no merge_index needs to be transmitted. Otherwise, the value of merge_index which is transmitted identifies the position of the candidate in the reduced list and its associated motion information which will be used for en-/decoding the current block X.

There is a conceptual similarity between the direct mode (of H.264/AVC) and our proposed block merging approach. Direct mode also infers the motion information for motion-compensated prediction (MCP) from previously decoded blocks. Thus, both algorithms are using motion information from neighboring blocks in order to reduce the bit rate required for transmitting motion parameters. However, for direct mode the motion information is derived independently for each reference picture. This means that for each reference picture (two in the HHI video codec), a motion predictor needs to be calculated from neighboring blocks for the current block to be en-/decoded. To derive a motion predictor in direct mode, up to four motion vectors of neighboring blocks are used, namely the above, left, above-left and above-right neighbors. The median of these vector components is used to calculate the motion predictor of the current block. For direct mode in H.264/AVC, also a flag is signaled for MCP partitions indicating whether the partition is coded in direct mode or not [1].

Despite this similarity, the two algorithms differ in the way they handle neighboring motion information. While the direct mode infers motion parameters from adjacent blocks, merging creates regions where all the blocks share the complete motion information. The creation of such regions is performed using only simple operations, such as comparing and copying the complete motion information from a neighbor. In comparison with this, the calculation of motion predictors in direct mode requires more computational complexity. Motion

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000120

Class	Sequence	BD rate CS1 [%]
B1 (1920x1080)	Kimono	-38.30
	ParkScene	-24.28
B2 (1920x1080)	Cactus	-29.22
	Basketball Drive	-35.97
	BQTerrace	-41.92
C (832x480)	Basketball Drill	-31.88
	BQMall	-29.71
	PartyScene	-28.09
	RaceHorses	-29.67
D (416x240)	Basketball Pass	-21.97
	BQSquare	-43.96
	BlowingBubbles	-23.60
	RaceHorses	-19.64
Average		**-30.63**

Table 1. Average BD rate savings (percentage) of the HHI codec relative to the H.264/AVC HP anchor at CS1 test conditions.

Class	Sequences	BD rate CS1 [%]	BD rate CS2 [%]
B1 (1920x1080)	Kimono	-2.42	-2.37
	ParkScene	-2.78	-2.88
B2 (1920x1080)	Cactus	-2.65	-2.28
	Basketball Drive	-2.93	-2.56
	BQTerrace	-3.45	-2.62
C (832x480)	Basketball Drill	-2.79	-2.58
	BQMall	-4.80	-4.47
	PartyScene	-2.22	-3.46
	RaceHorses	-3.08	-3.00
D (416x240)	Basketball Pass	-2.82	-2.11
	BQSquare	-2.41	-2.14
	BlowingBubbles	-2.10	-1.98
	RaceHorses	-2.81	-2.51
Average		**-2.87**	**-2.70**
E (1280x720)	Vidyo1	n/a	-4.50
	Vidyo3	n/a	-5.79
	Vidyo4	n/a	-4.32
Total Average		**-2.87**	**-3.10**

Table 2. Averaged BD rate savings (in percentage) of the HHI codec with block merging relative to the same codec with direct mode (instead of block merging).

predictors are calculated for all reference pictures individually while using up to four neighboring blocks, where additional operations are required to calculate the mean/median values, in case there is more than one neighboring motion vector available.

5. SIMULATION SETTINGS AND RESULTS

The codec used for performing our experiments is the HHI video codec as presented in section 2. The experiments were conducted by encoding all test sequences, as defined in the CfP [8]. Two constraint sets (CS) restricting the temporal coding structure were defined in the CfP as follows:

- CS1: hierarchical B picture coding structure [9] with a structural temporal delay not larger than 8 pictures,
- CS2: hierarchical P picture coding structure without picture reordering in the decoder.

Furthermore, the coding tree block has a fixed size of 64×64 pixels (for luma) and a maximum quadtree depth of 4, resulting in a minimum block size of 4×4. In order to obey CS1, inter-prediction with up to two hypotheses is employed within groups of 8 successive pictures in a hierarchical structure using 4 temporal layers. For CS2, inter prediction is restricted to one hypothesis from temporally preceding reference pictures. The intra period in CS1 is set depending on the number of frames per second for each sequence, whereas in CS2, only one intra picture at the beginning of each sequence has been utilized. Five quantization parameter (QP) values are chosen to obtain five RD points per sequence. The test sequences are separated into 4 classes according to their resolution and their content. For instance, the sequences of class E show typical videoconferencing scenes with static background. That is also why class E is only tested for CS2, the use case for low-delay applications. The other classes are characterized by different resolutions and a variety of content. With the exception of class E, all classes are tested for both CS1 and CS2. All presented bit rate savings are measured in terms of Bjøntegaard Delta (BD) rate [10].

First, the HHI proposal for HEVC (which incorporates block merging) has been evaluated in comparison to H.264/AVC high profile (HP) as an anchor [8]. Table 1 shows the bit rate savings of the HHI proposal relative to this anchor. The average gain for CS1 is around 30%.

The efficiency of block merging has been evaluated by comparing two versions of the HHI codec. The first version is given by the original HHI proposal for HEVC (including block merging), while the other version incorporates direct mode instead of block merging, as described in Section 2. The results of this comparison at CS1 and CS2 test conditions with the direct mode version as a reference are listed in Table 2. It can be observed that the gains of block merging in terms of average BD rate savings are around 3%, and that, to a large extent, those gains are independent of resolution or sequence type. However, there are several sequences where coding efficiency lies significantly above the average such as, e.g., all class E sequences, as shown in Table 2. These particularly favorable results for class E are mostly due to the static background of these sequences which can be easily represented by using a single merged region with zero motion vector at the corresponding block merging seed.

In Fig. 3, the RD curves of the sequence BQMall for CS1 are presented. The curves show the RD performance of three competing schemes: the HHI codec with block merging (the HHI proposal for HEVC), the HHI codec with direct mode,

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000121

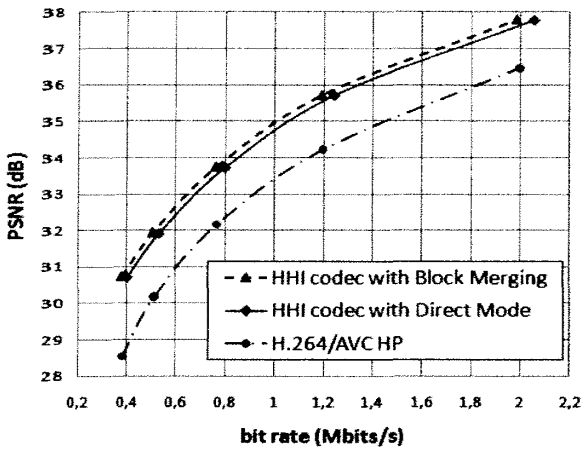


Fig. 3. RD performance plot comparing direct mode and block merging in the HHI video codec as well as H.264/AVC HP by using the test sequence BQMall at CS1 test conditions.

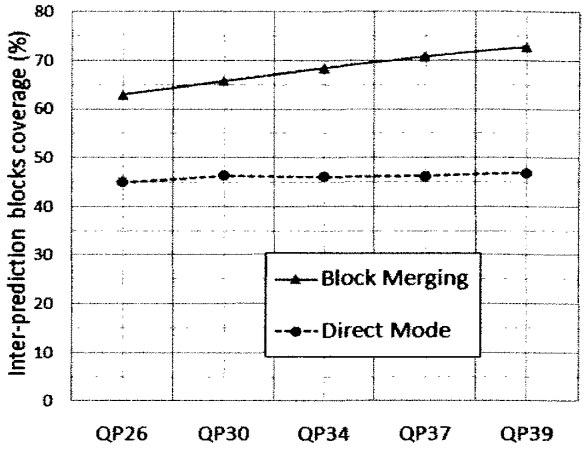


Fig. 4. Percentage coverage of MCP blocks using the direct mode and MCP blocks using block merging, both depending on the QP by using the sequence BQMall at CS1 test conditions.

and the H.264/AVC HP anchor scheme. For BQMall, the average BD rate saving gain of the HHI proposal relative to H.264/AVC HP is 29.7%. The average gain relative to the HHI codec with direct mode is 4.8% BD rate savings, where the individual gains for the different rate points are distributed more or less uniformly, which has also been observed for all other sequences.

Fig. 4 compares the percentage of picture samples covered by MCP blocks using direct mode with that using block merging. The curves were obtained by encoding the BQMall sequence at different QP values using both versions of the HHI codec. As can be seen from Fig. 4, the percentage of coverage for direct mode in comparison with that for block merging is much smaller. This behavior was consistently observed for the entire test set of sequences, which is an indication for block merging being selected more often due to its more favorable RD cost. Also, it can be observed that with larger QP values, the coverage increases for either method, although more distinctively for the case of block merging.

6. ILLUSTRATION OF THE RESULTS OBTAINED WITH BLOCK MERGING

In the following, details of encoded frames are further analyzed to demonstrate the motion modeling capabilities of block merging. In Figs. 5 and 6, white lines separate merged regions and grey lines indicate MCP leaf blocks of CTBs.

Fig. 5 shows a detail from a picture belonging to the BQMall sequence. In this part of the sequence, the background is static and only people in the foreground are moving. As can be seen from Fig. 5, the boundaries of the merged regions are clearly separating the moving persons from the static background.

Fig. 6 shows that block merging is effective also for more complex sequences. The corresponding Parkscene sequence was shot with a panning camera in front of a scene consisting of a park in the background and some tree branches in the foreground. The figure reveals distinct 'object planes' in the picture, all of which are moving differently such that each plane can be identified by a different set of block merging regions in Fig. 6. The tree branches correspond to the foreground, the colonnade belongs to a second plane, and the rest of the park can be attributed to the background.

As in the two examples examined above, natural video signals very often contain distinct moving objects, each with a different kind of motion. At the boundary of such objects, the field of motion parameters can be expected to exhibit very abrupt changes. These motion discontinuities can be modeled very efficiently by block merging, as it selects exactly one of the neighboring blocks to infer the motion information for the current block, leaving other candidates aside. In contrast to that, direct mode uses up to four spatially neighboring blocks to calculate an intermediate motion representation which, in the presence of distinct moving objects, may not be adequately representing the real motion in a given scene.

7. CONCLUSION AND FUTURE WORK

In this paper, block merging together with a quadtree-based block partitioning concept has been presented as an alternative to the conceptually similar direct mode for motion-compensated prediction. Simulation results show that the

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000122

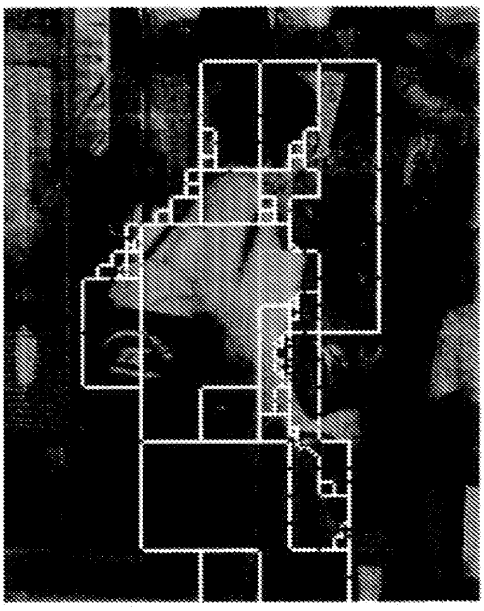


Fig. 5. Detail of sequence BQMall in Class C with a static background and moving persons in the foreground.

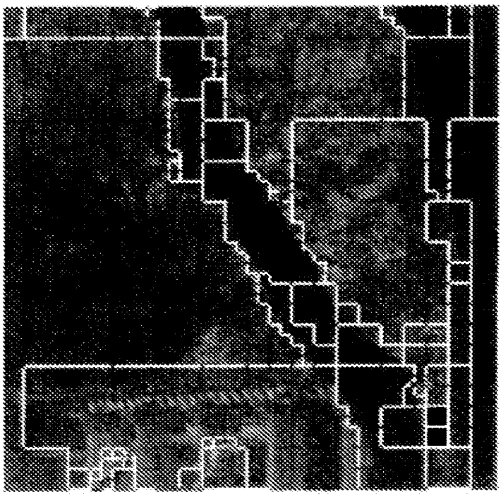


Fig. 6. Detail of sequence Parkscene in Class B with tree branches in the foreground and colonnades at the lower picture boundary, captured by a panning camera.

proposed merging algorithm better exploits spatiotemporal correlations in motion model parameters. Averaged BD rate savings of around 3% for a whole test set of video sequences have been achieved in comparison to the direct mode. Block merging not only yields better RD coding performance than the traditional direct mode, but also offers a computationally less demanding algorithm. These benefits were crucial for the proposed block merging algorithm to be adopted as part of the working draft of HEVC [11] which has been currently issued by the Joint Collaborative Team on Video Coding (JCT-VC). Further aspects of block merging within the HEVC test model are currently under investigation and further study [12].

8. REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 560–576, Jul. 2003.

[2] A. M. Tourapis, F. Wu, and S. Li, "Direct Mode Coding for Bipredicitve Slices in the H.264 Standard, *IEEE Trans. Ciruits and Syst. Video Technol.*, vol. 15, no. 1, pp. 119–126, Jan. 2005.

[3] M. Winken , S. Boße, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, D. Marpe, S. Oudin, M. Preiß, H. Schwarz, M. Siekmann, K. Sühring, T. Wiegand, "Description of video coding technology proposal by Fraunhofer HHI", Doc. *JCTVC-A116*, Dresden, Germany, Apr. 2010.

[4] D. Marpe, H. Schwarz, S. Boße, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, T. Nguyen, S. Oudin, M. Siekmann, K. Sühring, M. Winken, and T. Wiegand, "Video Compression Using Quadtrees, Leaf Merging and Novel Techniques for Motion Representation and Entropy Coding", *IEEE Trans. Ciruits and Syst. Video Technol.*, vol. 20, no. 12, Dec. 2010.

[5] R. Shukla, P. L. Dragotti, M. N. Do and M. Vetterli, "Rate-Distortion Optimized Tree-Structured Compression Algorithms for Piecewise Polynomial Images", *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 343–359, Mar. 2005.

[6] R. De Forni and D. S. Taubman, "On the benefits of leaf merging in quadtree motion models", *Proc. IEEE Int. Conf. Image Process.*, vol. 2., pp. 858–861, Sep. 2005.

[7] R. Mathew, D. S. Taubman, "Quad-Tree Motion Modeling with Leaf Merging", *IEEE Trans. Ciruits and Syst. Video Technol.*, vol. 20, no. 10, pp. 1331–1345, Oct. 2010.

[8] ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/16, "Joint Call for Proposals on Video Compression Technology," *WG11 Doc. N11113* and *ITU-T Q6/16 Doc. VCEG-AM91*, Kyoto, Jan. 2010.

[9] H. Schwarz, D. Marpe, T. Wiegand, "Hierarchical B pictures," *Joint Video Team, Doc. JVT-P014*, Poznan, Poland, July 2005.

[10] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD Curves, *ITU-T Q6/16 Doc. VCEG-M33*, April 2001.

[11] T. Wiegand, W.-J. Han, B. Bross, J. R. Ohm, G. J. Sullivan, "WD2: Working Draft 2 of High-Efficiency Video Coding" JCT-VC Meeting, Daegu, KR, Doc. *JCTVC-D503*, Output Document, Jan. 2011.

[12] J. Jung, B. Bross, P. Chen, W.-J. Han, "Summary report of Core Experiment 9: MV Coding and Skip/Merge operations" JCT-VC Meeting, Geneva, Switzerland, Doc. *JCTVC-E029*, Input Document, Jan. 2011.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000123

Overview of the High Efficiency Video Coding (HEVC) Standard

Gary J. Sullivan, *Fellow, IEEE*, Jens-Rainer Ohm, *Member, IEEE*, Woo-Jin Han, *Member, IEEE*, and Thomas Wiegand, *Fellow, IEEE*

***Abstract*—High Efficiency Video Coding (HEVC) is currently being prepared as the newest video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. The main goal of the HEVC standardization effort is to enable significantly improved compression performance relative to existing standards—in the range of 50% bit-rate reduction for equal perceptual video quality. This paper provides an overview of the technical features and characteristics of the HEVC standard.**

***Index Terms*—Advanced video coding (AVC), H.264, High Efficiency Video Coding (HEVC), Joint Collaborative Team on Video Coding (JCT-VC), Moving Picture Experts Group (MPEG), MPEG-4, standards, Video Coding Experts Group (VCEG), video compression.**

I. Introduction

THE High Efficiency Video Coding (HEVC) standard is the most recent joint video project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations, working together in a partnership known as the Joint Collaborative Team on Video Coding (JCT-VC) [1]. The first edition of the HEVC standard is expected to be finalized in January 2013, resulting in an aligned text that will be published by both ITU-T and ISO/IEC. Additional work is planned to extend the standard to support several additional application scenarios, including extended-range uses with enhanced precision and color format support, scalable video coding, and 3-D/stereo/multiview video coding. In ISO/IEC, the HEVC standard will become MPEG-H Part 2 (ISO/IEC 23008-2) and in ITU-T it is likely to become ITU-T Recommendation H.265.

Manuscript received May 25, 2012; revised August 22, 2012; accepted August 24, 2012. Date of publication October 2, 2012; date of current version January 8, 2013. This paper was recommended by Associate Editor H. Gharavi. (*Corresponding author: W.-J. Han.*)

G. J. Sullivan is with Microsoft Corporation, Redmond, WA 98052 USA (e-mail: garysull@microsoft.com).

J.-R. Ohm is with the Institute of Communication Engineering, RWTH Aachen University, Aachen 52056, Germany (e-mail: ohm@ient.rwth-aachen.de).

W.-J. Han is with the Department of Software Design and Management, Gachon University, Seongnam 461-701, Korea (e-mail: hurumi@gmail.com).

T. Wiegand is with the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin 10587, Germany, and also with the Berlin Institute of Technology, Berlin 10587, Germany (e-mail: twiegand@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2012.2221191

Video coding standards have evolved primarily through the development of the well-known ITU-T and ISO/IEC standards. The ITU-T produced H.261 [2] and H.263 [3], ISO/IEC produced MPEG-1 [4] and MPEG-4 Visual [5], and the two organizations jointly produced the H.262/MPEG-2 Video [6] and H.264/MPEG-4 Advanced Video Coding (AVC) [7] standards. The two standards that were jointly produced have had a particularly strong impact and have found their way into a wide variety of products that are increasingly prevalent in our daily lives. Throughout this evolution, continued efforts have been made to maximize compression capability and improve other characteristics such as data loss robustness, while considering the computational resources that were practical for use in products at the time of anticipated deployment of each standard.

The major video coding standard directly preceding the HEVC project was H.264/MPEG-4 AVC, which was initially developed in the period between 1999 and 2003, and then was extended in several important ways from 2003–2009. H.264/MPEG-4 AVC has been an enabling technology for digital video in almost every area that was not previously covered by H.262/MPEG-2 Video and has substantially displaced the older standard within its existing application domains. It is widely used for many applications, including broadcast of high definition (HD) TV signals over satellite, cable, and terrestrial transmission systems, video content acquisition and editing systems, camcorders, security applications, Internet and mobile network video, Blu-ray Discs, and real-time conversational applications such as video chat, video conferencing, and telepresence systems.

However, an increasing diversity of services, the growing popularity of HD video, and the emergence of beyond-HD formats (e.g., 4k×2k or 8k×4k resolution) are creating even stronger needs for coding efficiency superior to H.264/MPEG-4 AVC's capabilities. The need is even stronger when higher resolution is accompanied by stereo or multiview capture and display. Moreover, the traffic caused by video applications targeting mobile devices and tablet PCs, as well as the transmission needs for video-on-demand services, are imposing severe challenges on today's networks. An increased desire for higher quality and resolutions is also arising in mobile applications.

HEVC has been designed to address essentially all existing applications of H.264/MPEG-4 AVC and to particularly focus on two key issues: increased video resolution and increased use of parallel processing architectures. The syntax of HEVC

1051-8215/$31.00 © 2012 IEEE

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000124

is generic and should also be generally suited for other applications that are not specifically mentioned above.

As has been the case for all past ITU-T and ISO/IEC video coding standards, in HEVC only the bitstream structure and syntax is standardized, as well as constraints on the bitstream and its mapping for the generation of decoded pictures. The mapping is given by defining the semantic meaning of syntax elements and a decoding process such that every decoder conforming to the standard will produce the same output when given a bitstream that conforms to the constraints of the standard. This limitation of the scope of the standard permits maximal freedom to optimize implementations in a manner appropriate to specific applications (balancing compression quality, implementation cost, time to market, and other considerations). However, it provides no guarantees of end-to-end reproduction quality, as it allows even crude encoding techniques to be considered conforming.

To assist the industry community in learning how to use the standard, the standardization effort not only includes the development of a text specification document, but also reference software source code as an example of how HEVC video can be encoded and decoded. The draft reference software has been used as a research tool for the internal work of the committee during the design of the standard, and can also be used as a general research tool and as the basis of products. A standard test data suite is also being developed for testing conformance to the standard.

This paper is organized as follows. Section II highlights some key features of the HEVC coding design. Section III explains the high-level syntax and the overall structure of HEVC coded data. The HEVC coding technology is then described in greater detail in Section IV. Section V explains the profile, tier, and level design of HEVC. Since writing an overview of a technology as substantial as HEVC involves a significant amount of summarization, the reader is referred to [1] for any omitted details. The history of the HEVC standardization effort is discussed in Section VI.

II. HEVC Coding Design and Feature Highlights

The HEVC standard is designed to achieve multiple goals, including coding efficiency, ease of transport system integration and data loss resilience, as well as implementability using parallel processing architectures. The following subsections briefly describe the key elements of the design by which these goals are achieved, and the typical encoder operation that would generate a valid bitstream. More details about the associated syntax and the decoding process of the different elements are provided in Sections III and IV.

A. Video Coding Layer

The video coding layer of HEVC employs the same hybrid approach (inter-/intrapicture prediction and 2-D transform coding) used in all video compression standards since H.261. Fig. 1 depicts the block diagram of a hybrid video encoder, which could create a bitstream conforming to the HEVC standard.

An encoding algorithm producing an HEVC compliant bitstream would typically proceed as follows. Each picture is split into block-shaped regions, with the exact block partitioning being conveyed to the decoder. The first picture of a video sequence (and the first picture at each clean random access point into a video sequence) is coded using only intrapicture prediction (that uses some prediction of data spatially from region-to-region within the same picture, but has no dependence on other pictures). For all remaining pictures of a sequence or between random access points, interpicture temporally predictive coding modes are typically used for most blocks. The encoding process for interpicture prediction consists of choosing motion data comprising the selected reference picture and motion vector (MV) to be applied for predicting the samples of each block. The encoder and decoder generate identical interpicture prediction signals by applying motion compensation (MC) using the MV and mode decision data, which are transmitted as side information.

The residual signal of the intra- or interpicture prediction, which is the difference between the original block and its prediction, is transformed by a linear spatial transform. The transform coefficients are then scaled, quantized, entropy coded, and transmitted together with the prediction information.

The encoder duplicates the decoder processing loop (see gray-shaded boxes in Fig. 1) such that both will generate identical predictions for subsequent data. Therefore, the quantized transform coefficients are constructed by inverse scaling and are then inverse transformed to duplicate the decoded approximation of the residual signal. The residual is then added to the prediction, and the result of that addition may then be fed into one or two loop filters to smooth out artifacts induced by block-wise processing and quantization. The final picture representation (that is a duplicate of the output of the decoder) is stored in a decoded picture buffer to be used for the prediction of subsequent pictures. In general, the order of encoding or decoding processing of pictures often differs from the order in which they arrive from the source; necessitating a distinction between the decoding order (i.e., bitstream order) and the output order (i.e., display order) for a decoder.

Video material to be encoded by HEVC is generally expected to be input as progressive scan imagery (either due to the source video originating in that format or resulting from deinterlacing prior to encoding). No explicit coding features are present in the HEVC design to support the use of interlaced scanning, as interlaced scanning is no longer used for displays and is becoming substantially less common for distribution. However, a metadata syntax has been provided in HEVC to allow an encoder to indicate that interlace-scanned video has been sent by coding each field (i.e., the even or odd numbered lines of each video frame) of interlaced video as a separate picture or that it has been sent by coding each interlaced frame as an HEVC coded picture. This provides an efficient method of coding interlaced video without burdening decoders with a need to support a special decoding process for it.

In the following, the various features involved in hybrid video coding using HEVC are highlighted as follows.

1) *Coding tree units and coding tree block (CTB) structure:* The core of the coding layer in previous standards was

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000125

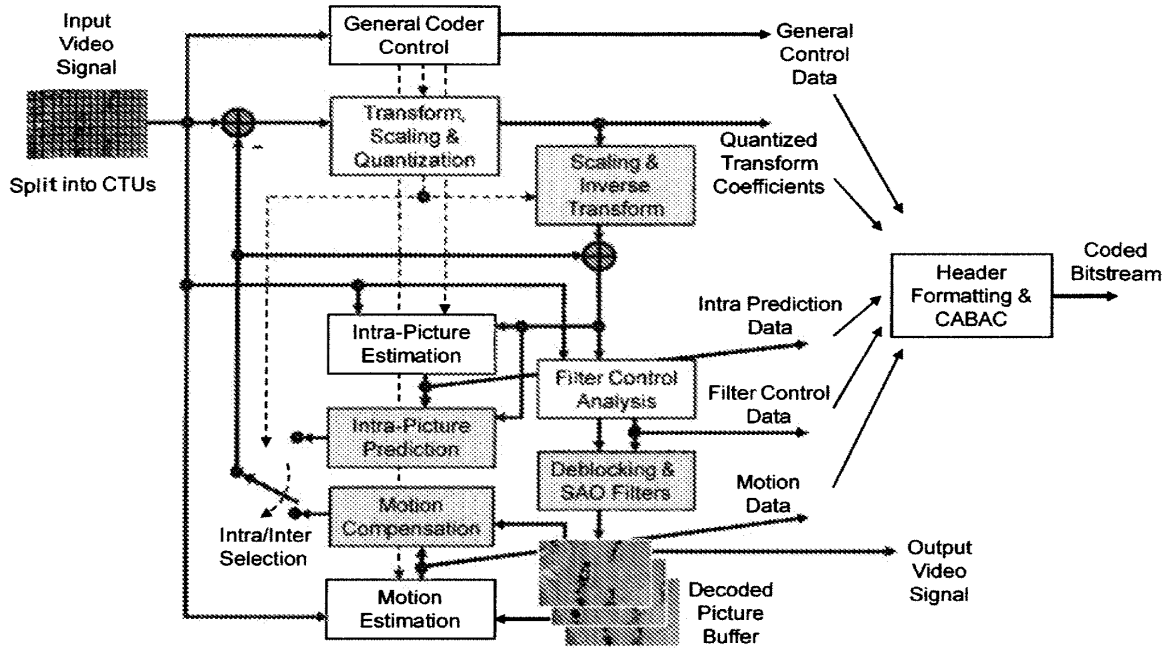


Fig. 1. Typical HEVC video encoder (with decoder modeling elements shaded in light gray).

the macroblock, containing a 16×16 block of luma samples and, in the usual case of 4:2:0 color sampling, two corresponding 8×8 blocks of chroma samples; whereas the analogous structure in HEVC is the coding tree unit (CTU), which has a size selected by the encoder and can be larger than a traditional macroblock. The CTU consists of a luma CTB and the corresponding chroma CTBs and syntax elements. The size $L \times L$ of a luma CTB can be chosen as $L = 16$, 32, or 64 samples, with the larger sizes typically enabling better compression. HEVC then supports a partitioning of the CTBs into smaller blocks using a tree structure and quadtree-like signaling [8].

2) *Coding units (CUs) and coding blocks (CBs):* The quadtree syntax of the CTU specifies the size and positions of its luma and chroma CBs. The root of the quadtree is associated with the CTU. Hence, the size of the luma CTB is the largest supported size for a luma CB. The splitting of a CTU into luma and chroma CBs is signaled jointly. One luma CB and ordinarily two chroma CBs, together with associated syntax, form a coding unit (CU). A CTB may contain only one CU or may be split to form multiple CUs, and each CU has an associated partitioning into prediction units (PUs) and a tree of transform units (TUs).
3) *Prediction units and prediction blocks (PBs):* The decision whether to code a picture area using interpicture or intrapicture prediction is made at the CU level. A PU partitioning structure has its root at the CU level. Depending on the basic prediction-type decision, the luma and chroma CBs can then be further split in size and predicted from luma and chroma prediction blocks (PBs). HEVC supports variable PB sizes from 64×64 down to 4×4 samples.
4) *TUs and transform blocks*: The prediction residual is coded using block transforms. A TU tree structure has its root at the CU level. The luma CB residual may be identical to the luma transform block (TB) or may be further split into smaller luma TBs. The same applies to the chroma TBs. Integer basis functions similar to those of a discrete cosine transform (DCT) are defined for the square TB sizes 4×4, 8×8, 16×16, and 32×32. For the 4×4 transform of luma intrapicture prediction residuals, an integer transform derived from a form of discrete sine transform (DST) is alternatively specified.
5) *Motion vector signaling:* Advanced motion vector prediction (AMVP) is used, including derivation of several most probable candidates based on data from adjacent PBs and the reference picture. A merge mode for MV coding can also be used, allowing the inheritance of MVs from temporally or spatially neighboring PBs. Moreover, compared to H.264/MPEG-4 AVC, improved skipped and direct motion inference are also specified.
6) *Motion compensation:* Quarter-sample precision is used for the MVs, and 7-tap or 8-tap filters are used for interpolation of fractional-sample positions (compared to six-tap filtering of half-sample positions followed by linear interpolation for quarter-sample positions in

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000126

H.264/MPEG-4 AVC). Similar to H.264/MPEG-4 AVC, multiple reference pictures are used. For each PB, either one or two motion vectors can be transmitted, resulting either in unipredictive or bipredictive coding, respectively. As in H.264/MPEG-4 AVC, a scaling and offset operation may be applied to the prediction signal(s) in a manner known as weighted prediction.

7) *Intrapicture prediction:* The decoded boundary samples of adjacent blocks are used as reference data for spatial prediction in regions where interpicture prediction is not performed. Intrapicture prediction supports 33 directional modes (compared to eight such modes in H.264/MPEG-4 AVC), plus planar (surface fitting) and DC (flat) prediction modes. The selected intrapicture prediction modes are encoded by deriving most probable modes (e.g., prediction directions) based on those of previously decoded neighboring PBs.
8) *Quantization control:* As in H.264/MPEG-4 AVC, uniform reconstruction quantization (URQ) is used in HEVC, with quantization scaling matrices supported for the various transform block sizes.
9) *Entropy coding:* Context adaptive binary arithmetic coding (CABAC) is used for entropy coding. This is similar to the CABAC scheme in H.264/MPEG-4 AVC, but has undergone several improvements to improve its throughput speed (especially for parallel-processing architectures) and its compression performance, and to reduce its context memory requirements.
10) *In-loop deblocking filtering:* A deblocking filter similar to the one used in H.264/MPEG-4 AVC is operated within the interpicture prediction loop. However, the design is simplified in regard to its decision-making and filtering processes, and is made more friendly to parallel processing.
11) *Sample adaptive offset (SAO):* A nonlinear amplitude mapping is introduced within the interpicture prediction loop after the deblocking filter. Its goal is to better reconstruct the original signal amplitudes by using a look-up table that is described by a few additional parameters that can be determined by histogram analysis at the encoder side.

B. High-Level Syntax Architecture

A number of design aspects new to the HEVC standard improve flexibility for operation over a variety of applications and network environments and improve robustness to data losses. However, the high-level syntax architecture used in the H.264/MPEG-4 AVC standard has generally been retained, including the following features.

1) *Parameter set structure:* Parameter sets contain information that can be shared for the decoding of several regions of the decoded video. The parameter set structure provides a robust mechanism for conveying data that are essential to the decoding process. The concepts of sequence and picture parameter sets from H.264/MPEG-4 AVC are augmented by a new video parameter set (VPS) structure.
2) *NAL unit syntax structure*: Each syntax structure is placed into a logical data packet called a network abstraction layer (NAL) unit. Using the content of a two-byte NAL unit header, it is possible to readily identify the purpose of the associated payload data.
3) *Slices*: A slice is a data structure that can be decoded independently from other slices of the same picture, in terms of entropy coding, signal prediction, and residual signal reconstruction. A slice can either be an entire picture or a region of a picture. One of the main purposes of slices is resynchronization in the event of data losses. In the case of packetized transmission, the maximum number of payload bits within a slice is typically restricted, and the number of CTUs in the slice is often varied to minimize the packetization overhead while keeping the size of each packet within this bound.
4) *Supplemental enhancement information (SEI) and video usability information (VUI) metadata*: The syntax includes support for various types of metadata known as SEI and VUI. Such data provide information about the timing of the video pictures, the proper interpretation of the color space used in the video signal, 3-D stereoscopic frame packing information, other display hint information, and so on.

C. Parallel Decoding Syntax and Modified Slice Structuring

Finally, four new features are introduced in the HEVC standard to enhance the parallel processing capability or modify the structuring of slice data for packetization purposes. Each of them may have benefits in particular application contexts, and it is generally up to the implementer of an encoder or decoder to determine whether and how to take advantage of these features.

1) *Tiles*: The option to partition a picture into rectangular regions called tiles has been specified. The main purpose of tiles is to increase the capability for parallel processing rather than provide error resilience. Tiles are independently decodable regions of a picture that are encoded with some shared header information. Tiles can additionally be used for the purpose of spatial random access to local regions of video pictures. A typical tile configuration of a picture consists of segmenting the picture into rectangular regions with approximately equal numbers of CTUs in each tile. Tiles provide parallelism at a more coarse level of granularity (picture/subpicture), and no sophisticated synchronization of threads is necessary for their use.
2) *Wavefront parallel processing*: When wavefront parallel processing (WPP) is enabled, a slice is divided into rows of CTUs. The first row is processed in an ordinary way, the second row can begin to be processed after only two CTUs have been processed in the first row, the third row can begin to be processed after only two CTUs have been processed in the second row, and so on. The context models of the entropy coder in each row are inferred from those in the preceding row with a two-CTU processing lag. WPP provides a form of processing parallelism at a rather fine level of

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000127

granularity, i.e., within a slice. WPP may often provide better compression performance than tiles (and avoid some visual artifacts that may be induced by using tiles).

3) *Dependent slice segments*: A structure called a dependent slice segment allows data associated with a particular wavefront entry point or tile to be carried in a separate NAL unit, and thus potentially makes that data available to a system for fragmented packetization with lower latency than if it were all coded together in one slice. A dependent slice segment for a wavefront entry point can only be decoded after at least part of the decoding process of another slice segment has been performed. Dependent slice segments are mainly useful in low-delay encoding, where other parallel tools might penalize compression performance.

In the following two sections, a more detailed description of the key features is given.

III. High-Level Syntax

The high-level syntax of HEVC contains numerous elements that have been inherited from the NAL of H.264/MPEG-4 AVC. The NAL provides the ability to map the video coding layer (VCL) data that represent the content of the pictures onto various transport layers, including RTP/IP, ISO MP4, and H.222.0/MPEG-2 Systems, and provides a framework for packet loss resilience. For general concepts of the NAL design such as NAL units, parameter sets, access units, the byte stream format, and packetized formatting, please refer to [9]–[11].

NAL units are classified into VCL and non-VCL NAL units according to whether they contain coded pictures or other associated data, respectively. In the HEVC standard, several VCL NAL unit types identifying categories of pictures for decoder initialization and random-access purposes are included. Table I lists the NAL unit types and their associated meanings and type classes in the HEVC standard.

The following subsections present a description of the new capabilities supported by the high-level syntax.

A. Random Access and Bitstream Splicing Features

The new design supports special features to enable random access and bitstream splicing. In H.264/MPEG-4 AVC, a bitstream must always start with an IDR access unit. An IDR access unit contains an independently coded picture—i.e., a coded picture that can be decoded without decoding any previous pictures in the NAL unit stream. The presence of an IDR access unit indicates that no subsequent picture in the bitstream will require reference to pictures prior to the picture that it contains in order to be decoded. The IDR picture is used within a coding structure known as a closed GOP (in which GOP stands for group of pictures).

The new clean random access (CRA) picture syntax specifies the use of an independently coded picture at the location of a random access point (RAP), i.e., a location in a bitstream at which a decoder can begin successfully decoding pictures without needing to decode any pictures that appeared earlier in the bitstream, which supports an efficient temporal coding order known as open GOP operation. Good support of random access is critical for enabling channel switching, seek operations, and dynamic streaming services. Some pictures that follow a CRA picture in decoding order and precede it in display order may contain interpicture prediction references to pictures that are not available at the decoder. These nondecodable pictures must therefore be discarded by a decoder that starts its decoding process at a CRA point. For this purpose, such nondecodable pictures are identified as random access skipped leading (RASL) pictures. The location of splice points from different original coded bitstreams can be indicated by broken link access (BLA) pictures. A bitstream splicing operation can be performed by simply changing the NAL unit type of a CRA picture in one bitstream to the value that indicates a BLA picture and concatenating the new bitstream at the position of a RAP picture in the other bitstream. A RAP picture may be an IDR, CRA, or BLA picture, and both CRA and BLA pictures may be followed by RASL pictures in the bitstream (depending on the particular value of the NAL unit type used for a BLA picture). Any RASL pictures associated with a BLA picture must always be discarded by the decoder, as they may contain references to pictures that are not actually present in the bitstream due to a splicing operation. The other type of picture that can follow a RAP picture in decoding order and precede it in output order is the random access decodable leading (RADL) picture, which cannot contain references to any pictures that precede the RAP picture in decoding order. RASL and RADL pictures are collectively referred to as leading pictures (LPs). Pictures that follow a RAP picture in both decoding order and output order, which are known as trailing pictures, cannot contain references to LPs for interpicture prediction.

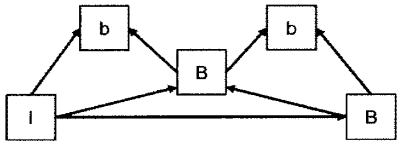
TABLE I
NAL Unit Types, Meanings, and Type Classes

Type	Meaning	Class
0, 1	Slice segment of ordinary trailing picture	VCL
2, 3	Slice segment of TSA picture	VCL
4, 5	Slice segment of STSA picture	VCL
6, 7	Slice segment of RADL picture	VCL
8, 9	Slice segment of RASL picture	VCL
10–15	Reserved for future use	VCL
16–18	Slice segment of BLA picture	VCL
19, 20	Slice segment of IDR picture	VCL
21	Slice segment of CRA picture	VCL
22–31	Reserved for future use	VCL
32	Video parameter set (VPS)	non-VCL
33	Sequence parameter set (SPS)	non-VCL
34	Picture parameter set (PPS)	non-VCL
35	Access unit delimiter	non-VCL
36	End of sequence	non-VCL
37	End of bitstream	non-VCL
38	Filler data	non-VCL
39, 40	SEI messages	non-VCL
41–47	Reserved for future use	non-VCL
48–63	Unspecified (available for system use)	non-VCL

B. Temporal Sublayering Support

Similar to the temporal scalability feature in the H.264/MPEG-4 AVC scalable video coding (SVC) extension [12],

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000128



POC	0	1	2	3	4
Decoding order	1	4	3	5	2
RPS	–	[0,2]	[0,4]	[2,4]	[0]

Fig. 2. Example of a temporal prediction structure and the POC values, decoding order, and RPS content for each picture.

HEVC specifies a temporal identifier in the NAL unit header, which indicates a level in a hierarchical temporal prediction structure. This was introduced to achieve temporal scalability without the need to parse parts of the bitstream other than the NAL unit header.

Under certain circumstances, the number of decoded temporal sublayers can be adjusted during the decoding process of one coded video sequence. The location of a point in the bitstream at which sublayer switching is possible to begin decoding some higher temporal layers can be indicated by the presence of temporal sublayer access (TSA) pictures and stepwise TSA (STSA) pictures. At the location of a TSA picture, it is possible to switch from decoding a lower temporal sublayer to decoding any higher temporal sublayer, and at the location of an STSA picture, it is possible to switch from decoding a lower temporal sublayer to decoding only one particular higher temporal sublayer (but not the further layers above that, unless they also contain STSA or TSA pictures).

C. Additional Parameter Sets

The VPS has been added as metadata to describe the overall characteristics of coded video sequences, including the dependences between temporal sublayers. The primary purpose of this is to enable the compatible extensibility of the standard in terms of signaling at the systems layer, e.g., when the base layer of a future extended scalable or multiview bitstream would need to be decodable by a legacy decoder, but for which additional information about the bitstream structure that is only relevant for the advanced decoder would be ignored.

D. Reference Picture Sets and Reference Picture Lists

For multiple-reference picture management, a particular set of previously decoded pictures needs to be present in the decoded picture buffer (DPB) for the decoding of the remainder of the pictures in the bitstream. To identify these pictures, a list of picture order count (POC) identifiers is transmitted in each slice header. The set of retained reference pictures is called the reference picture set (RPS). Fig. 2 shows POC values, decoding order, and RPSs for an example temporal prediction structure.

As in H.264/MPEG-4 AVC, there are two lists that are constructed as lists of pictures in the DPB, and these are called reference picture list 0 and list 1. An index called a reference picture index is used to identify a particular picture in one of these lists. For unipredication, a picture can be selected from either of these lists. For biprediction, two pictures are selected—one from each list. When a list contains only one picture, the reference picture index implicitly has the value 0 and does not need to be transmitted in the bitstream.

The high-level syntax for identifying the RPS and establishing the reference picture lists for interpicture prediction is more robust to data losses than in the prior H.264/MPEG-4 AVC design, and is more amenable to such operations as random access and trick mode operation (e.g., fast-forward, smooth rewind, seeking, and adaptive bitstream switching). A key aspect of this improvement is that the syntax is more explicit, rather than depending on inferences from the stored internal state of the decoding process as it decodes the bitstream picture by picture. Moreover, the associated syntax for these aspects of the design is actually simpler than it had been for H.264/MPEG-4 AVC.

IV. HEVC Video Coding Techniques

As in all prior ITU-T and ISO/IEC JTC 1 video coding standards since H.261 [2], the HEVC design follows the classic block-based hybrid video coding approach (as depicted in Fig. 1). The basic source-coding algorithm is a hybrid of interpicture prediction to exploit temporal statistical dependences, intrapicture prediction to exploit spatial statistical dependences, and transform coding of the prediction residual signals to further exploit spatial statistical dependences. There is no single coding element in the HEVC design that provides the majority of its significant improvement in compression efficiency in relation to prior video coding standards. It is, rather, a plurality of smaller improvements that add up to the significant gain.

A. Sampled Representation of Pictures

For representing color video signals, HEVC typically uses a tristimulus YCbCr color space with 4:2:0 sampling (although extension to other sampling formats is straightforward, and is planned to be defined in a subsequent version). This separates a color representation into three components called Y, Cb, and Cr. The Y component is also called luma, and represents brightness. The two chroma components Cb and Cr represent the extent to which the color deviates from gray toward blue and red, respectively. Because the human visual system is more sensitive to luma than chroma, the 4:2:0 sampling structure is typically used, in which each chroma component has one fourth of the number of samples of the luma component (half the number of samples in both the horizontal and vertical dimensions). Each sample for each component is typically represented with 8 or 10 b of precision, and the 8-b case is the more typical one. In the remainder of this paper, we focus our attention on the typical use: YCbCr components with 4:2:0 sampling and 8 b per sample for the representation of the encoded input and decoded output video signal.

The video pictures are typically progressively sampled with rectangular picture sizes $W \times H$, where W is the width and

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000129

H is the height of the picture in terms of luma samples. Each chroma component array, with 4:2:0 sampling, is then $W/2 \times H/2$. Given such a video signal, the HEVC syntax partitions the pictures further as described follows.

B. Division of the Picture into Coding Tree Units

A picture is partitioned into coding tree units (CTUs), which each contain luma CTBs and chroma CTBs. A luma CTB covers a rectangular picture area of $L \times L$ samples of the luma component and the corresponding chroma CTBs cover each $L/2 \times L/2$ samples of each of the two chroma components. The value of L may be equal to 16, 32, or 64 as determined by an encoded syntax element specified in the SPS. Compared with the traditional macroblock using a fixed array size of 16×16 luma samples, as used by all previous ITU-T and ISO/IEC JTC 1 video coding standards since H.261 (that was standardized in 1990), HEVC supports variable-size CTBs selected according to needs of encoders in terms of memory and computational requirements. The support of larger CTBs than in previous standards is particularly beneficial when encoding high-resolution video content. The luma CTB and the two chroma CTBs together with the associated syntax form a CTU. The CTU is the basic processing unit used in the standard to specify the decoding process.

C. Division of the CTB into CBs

The blocks specified as luma and chroma CTBs can be directly used as CBs or can be further partitioned into multiple CBs. Partitioning is achieved using tree structures. The tree partitioning in HEVC is generally applied simultaneously to both luma and chroma, although exceptions apply when certain minimum sizes are reached for chroma.

The CTU contains a quadtree syntax that allows for splitting the CBs to a selected appropriate size based on the signal characteristics of the region that is covered by the CTB. The quadtree splitting process can be iterated until the size for a luma CB reaches a minimum allowed luma CB size that is selected by the encoder using syntax in the SPS and is always 8×8 or larger (in units of luma samples).

The boundaries of the picture are defined in units of the minimum allowed luma CB size. As a result, at the right and bottom edges of the picture, some CTUs may cover regions that are partly outside the boundaries of the picture. This condition is detected by the decoder, and the CTU quadtree is implicitly split as necessary to reduce the CB size to the point where the entire CB will fit into the picture.

D. PBs and PUs

The prediction mode for the CU is signaled as being intra or inter, according to whether it uses intrapicture (spatial) prediction or interpicture (temporal) prediction.

When the prediction mode is signaled as intra, the PB size, which is the block size at which the intrapicture prediction mode is established is the same as the CB size for all block sizes except for the smallest CB size that is allowed in the bitstream. For the latter case, a flag is present that indicates whether the CB is split into four PB quadrants that each have their own intrapicture prediction mode. The reason for allowing this split is to enable distinct intrapicture prediction mode selections for blocks as small as 4×4 in size. When the luma intrapicture prediction operates with 4×4 blocks, the chroma intrapicture prediction also uses 4×4 blocks (each covering the same picture region as four 4×4 luma blocks). The actual region size at which the intrapicture prediction operates (which is distinct from the PB size, at which the intrapicture prediction mode is established) depends on the residual coding partitioning that is described as follows.

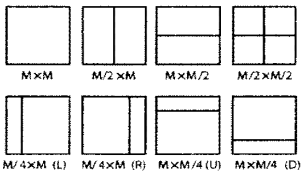


Fig. 3. Modes for splitting a CB into PBs, subject to certain size constraints. For intrapicture-predicted CBs, only $M \times M$ and $M/2 \times M/2$ are supported.

When the prediction mode is signaled as inter, it is specified whether the luma and chroma CBs are split into one, two, or four PBs. The splitting into four PBs is allowed only when the CB size is equal to the minimum allowed CB size, using an equivalent type of splitting as could otherwise be performed at the CB level of the design rather than at the PB level. When a CB is split into four PBs, each PB covers a quadrant of the CB. When a CB is split into two PBs, six types of this splitting are possible. The partitioning possibilities for interpicture-predicted CBs are depicted in Fig. 3. The upper partitions illustrate the cases of not splitting the CB of size $M \times M$, of splitting the CB into two PBs of size $M \times M/2$ or $M/2 \times M$, or splitting it into four PBs of size $M/2 \times M/2$. The lower four partition types in Fig. 3 are referred to as asymmetric motion partitioning (AMP), and are only allowed when M is 16 or larger for luma. One PB of the asymmetric partition has the height or width $M/4$ and width or height M, respectively, and the other PB fills the rest of the CB by having a height or width of $3M/4$ and width or height M. Each interpicture-predicted PB is assigned one or two motion vectors and reference picture indices. To minimize worst-case memory bandwidth, PBs of luma size 4×4 are not allowed for interpicture prediction, and PBs of luma sizes 4×8 and 8×4 are restricted to unipredictive coding. The interpicture prediction process is further described as follows.

The luma and chroma PBs, together with the associated prediction syntax, form the PU.

E. Tree-Structured Partitioning Into Transform Blocks and Units

For residual coding, a CB can be recursively partitioned into transform blocks (TBs). The partitioning is signaled by a residual quadtree.

Only square CB and TB partitioning is specified, where a block can be recursively split into quadrants, as illustrated in Fig. 4. For a given luma CB of size $M \times M$, a flag signals whether it is split into four blocks of size $M/2 \times M/2$. If

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000130

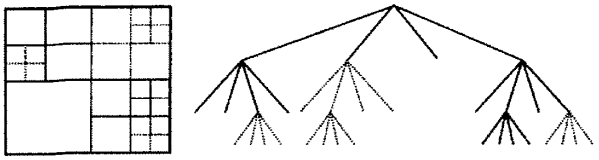


Fig. 4. Subdivision of a CTB into CBs [and transform block (TBs)]. Solid lines indicate CB boundaries and dotted lines indicate TB boundaries. (a) CTB with its partitioning. (b) Corresponding quadtree.

further splitting is possible, as signaled by a maximum depth of the residual quadtree indicated in the SPS, each quadrant is assigned a flag that indicates whether it is split into four quadrants. The leaf node blocks resulting from the residual quadtree are the transform blocks that are further processed by transform coding. The encoder indicates the maximum and minimum luma TB sizes that it will use. Splitting is implicit when the CB size is larger than the maximum TB size. Not splitting is implicit when splitting would result in a luma TB size smaller than the indicated minimum. The chroma TB size is half the luma TB size in each dimension, except when the luma TB size is 4×4, in which case a single 4×4 chroma TB is used for the region covered by four 4×4 luma TBs. In the case of intrapicture-predicted CUs, the decoded samples of the nearest-neighboring TBs (within or outside the CB) are used as reference data for intrapicture prediction.

In contrast to previous standards, the HEVC design allows a TB to span across multiple PBs for interpicture-predicted CUs to maximize the potential coding efficiency benefits of the quadtree-structured TB partitioning.

F. Slices and Tiles

Slices are a sequence of CTUs that are processed in the order of a raster scan. A picture may be split into one or several slices as shown in Fig. 5(a) so that a picture is a collection of one or more slices. Slices are self-contained in the sense that, given the availability of the active sequence and picture parameter sets, their syntax elements can be parsed from the bitstream and the values of the samples in the area of the picture that the slice represents can be correctly decoded (except with regard to the effects of in-loop filtering near the edges of the slice) without the use of any data from other slices in the same picture. This means that prediction within the picture (e.g., intrapicture spatial signal prediction or prediction of motion vectors) is not performed across slice boundaries. Some information from other slices may, however, be needed to apply the in-loop filtering across slice boundaries. Each slice can be coded using different coding types as follows.

1) *I slice:* A slice in which all CUs of the slice are coded using only intrapicture prediction.
2) *P slice:* In addition to the coding types of an I slice, some CUs of a P slice can also be coded using interpicture prediction with at most one motion-compensated prediction signal per PB (i.e., uniprediction). P slices only use reference picture list 0.
3) *B slice:* In addition to the coding types available in a P slice, some CUs of the B slice can also be coded using interpicture prediction with at most two motion-compensated prediction signals per PB (i.e., biprediction). B slices use both reference picture list 0 and list 1.

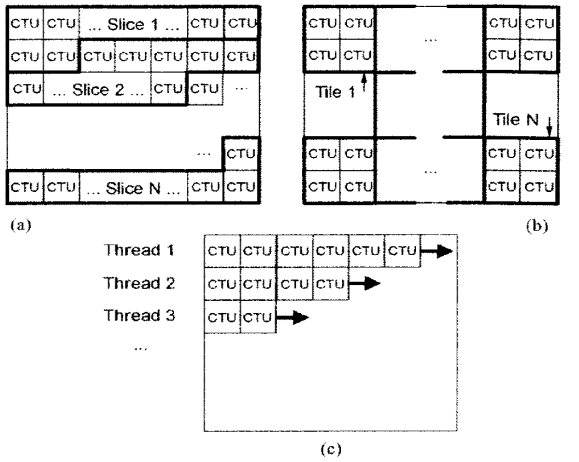


Fig. 5. Subdivision of a picture into (a) slices and (b) tiles. (c) Illustration of wavefront parallel processing.

The main purpose of slices is resynchronization after data losses. Furthermore, slices are often restricted to use a maximum number of bits, e.g., for packetized transmission. Therefore, slices may often contain a highly varying number of CTUs per slice in a manner dependent on the activity in the video scene. In addition to slices, HEVC also defines tiles, which are self-contained and independently decodable rectangular regions of the picture. The main purpose of tiles is to enable the use of parallel processing architectures for encoding and decoding. Multiple tiles may share header information by being contained in the same slice. Alternatively, a single tile may contain multiple slices. A tile consists of a rectangular arranged group of CTUs (typically, but not necessarily, with all of them containing about the same number of CTUs), as shown in Fig. 5(b).

To assist with the granularity of data packetization, dependent slices are additionally defined. Finally, with WPP, a slice is divided into rows of CTUs. The decoding of each row can be begun as soon a few decisions that are needed for prediction and adaptation of the entropy coder have been made in the preceding row. This supports parallel processing of rows of CTUs by using several processing threads in the encoder or decoder (or both). An example is shown in Fig. 5(c). For design simplicity, WPP is not allowed to be used in combination with tiles (although these features could, in principle, work properly together).

G. Intrapicture Prediction

Intrapicture prediction operates according to the TB size, and previously decoded boundary samples from spatially neighboring TBs are used to form the prediction signal. Directional prediction with 33 different directional orientations is defined for (square) TB sizes from 4×4 up to 32×32. The

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000131

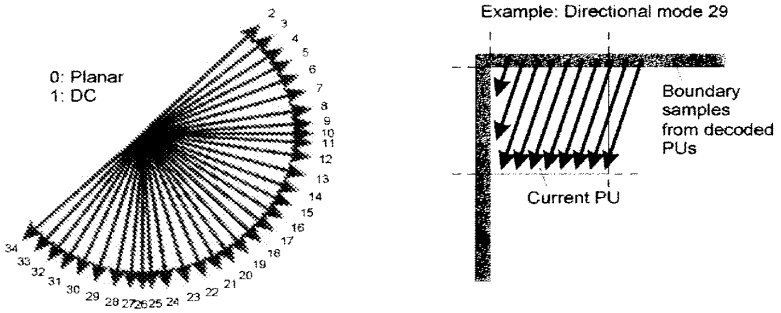


Fig. 6. Modes and directional orientations for intrapicture prediction.

possible prediction directions are shown in Fig. 6. Alternatively, planar prediction (assuming an amplitude surface with a horizontal and vertical slope derived from the boundaries) and DC prediction (a flat surface with a value matching the mean value of the boundary samples) can also be used. For chroma, the horizontal, vertical, planar, and DC prediction modes can be explicitly signaled, or the chroma prediction mode can be indicated to be the same as the luma prediction mode (and, as a special case to avoid redundant signaling, when one of the first four choices is indicated and is the same as the luma prediction mode, the Intra_Angular[34] mode is applied instead).

Each CB can be coded by one of several coding types, depending on the slice type. Similar to H.264/MPEG-4 AVC, intrapicture predictive coding is supported in all slice types. HEVC supports various intrapicture predictive coding methods referred to as Intra_Angular, Intra_Planar, and Intra_DC. The following subsections present a brief further explanation of these and several techniques to be applied in common.

1) *PB Partitioning:* An intrapicture-predicted CB of size $M \times M$ may have one of two types of PB partitions referred to as PART_2N×2N and PART_N×N, the first of which indicates that the CB is not split and the second indicates that the CB is split into four equal-sized PBs. (Conceptually, in this notation, $N = M/2$.) However, it is possible to represent the same regions that would be specified by four PBs by using four smaller CBs when the size of the current CB is larger than the minimum CU size. Thus, the HEVC design only allows the partitioning type PART_N×N to be used when the current CB size is equal to the minimum CU size. This means that the PB size is always equal to the CB size when the CB is coded using an intrapicture prediction mode and the CB size is not equal to the minimum CU size. Although the intrapicture prediction mode is established at the PB level, the actual prediction process operates separately for each TB.

2) *Intra_Angular Prediction:* Spatial-domain intrapicture prediction has previously been successfully used in H.264/MPEG-4 AVC. The intrapicture prediction of HEVC similarly operates in the spatial domain, but is extended significantly—mainly due to the increased size of the TB and an increased number of selectable prediction directions. Compared to the eight prediction directions of H.264/MPEG-4 AVC, HEVC supports a total of 33 prediction directions, denoted as Intra_Angular[k], where k is a mode number from 2 to 34. The angles are intentionally designed to provide denser coverage for near-horizontal and near-vertical angles and coarser coverage for near-diagonal angles to reflect the observed statistical prevalence of the angles and the effectiveness of the signal prediction processing.

When using an Intra_Angular mode, each TB is predicted directionally from spatially neighboring samples that are reconstructed (but not yet filtered by the in-loop filters) before being used for this prediction. For a TB of size $N \times N$, a total of $4N+1$ spatially neighboring samples may be used for the prediction, as shown in Fig. 6. When available from preceding decoding operations, samples from lower left TBs can be used for prediction in HEVC in addition to samples from TBs at the left, above, and above right of the current TB.

The prediction process of the Intra_Angular modes can involve extrapolating samples from the projected reference sample location according to a given directionality. To remove the need for sample-by-sample switching between reference row and column buffers, for Intra_Angular[k] with k in the range of 2–17, the samples located in the above row are projected as additional samples located in the left column; and with k in the range of 18–34, the samples located at the left column are projected as samples located in the above row.

To improve the intrapicture prediction accuracy, the projected reference sample location is computed with 1/32 sample accuracy. Bilinear interpolation is used to obtain the value of the projected reference sample using two closest reference samples located at integer positions.

The prediction process of the Intra_Angular modes is consistent across all block sizes and prediction directions, whereas H.264/MPEG-4 AVC uses different methods for its supported block sizes of 4×4, 8×8, and 16×16. This design consistency is especially desirable since HEVC supports a greater variety of TB sizes and a significantly increased number of prediction directions compared to H.264/MPEG-4 AVC.

3) *Intra_Planar and Intra_DC Prediction:* In addition to Intra_Angular prediction that targets regions with strong directional edges, HEVC supports two alternative prediction methods, Intra_Planar and Intra_DC, for which similar modes were specified in H.264/MPEG-4 AVC. While Intra_DC prediction uses an average value of reference samples for the prediction, average values of two linear predictions using four corner reference samples are used in Intra_Planar prediction

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000132

to prevent discontinuities along the block boundaries. The Intra_Planar prediction mode is supported at all block sizes in HEVC, while H.264/MPEG-4 AVC supports plane prediction only when the luma PB size is 16×16, and its plane prediction operates somewhat differently from the planar prediction in HEVC.

4) *Reference Sample Smoothing:* In HEVC, the reference samples used for the intrapicture prediction are sometimes filtered by a three-tap [1 2 1]/4 smoothing filter in a manner similar to what was used for 8×8 intrapicture prediction in H.264/MPEG-4 AVC. HEVC applies smoothing operations more adaptively, according to the directionality, the amount of detected discontinuity, and the block size. As in H.264/MPEG-4 AVC, the smoothing filter is not applied for 4×4 blocks. For 8×8 blocks, only the diagonal directions, Intra_Angular[k] with k = 2, 18, or 34, use the reference sample smoothing. For 16×16 blocks, the reference samples are filtered for most directions except the near-horizontal and near-vertical directions, k in the range of 9–11 and 25–27. For 32×32 blocks, all directions except the exactly horizontal (k = 10) and exactly vertical (k = 26) directions use the smoothing filter, and when the amount of detected discontinuity exceeds a threshold, bilinear interpolation from three neighboring region samples is applied to form a smooth prediction.

The Intra_Planar mode also uses the smoothing filter when the block size is greater than or equal to 8×8, and the smoothing is not used (or useful) for the Intra_DC case.

5) *Boundary Value Smoothing:* To remove discontinuities along block boundaries, in three modes, Intra_DC (mode 1) and Intra_Angular[k] with k = 10 or 26 (exactly horizontal or exactly vertical), the boundary samples inside the TB are replaced by filtered values when the TB size is smaller than 32 × 32. For Intra_DC mode, both the first row and column of samples in the TB are replaced by the output of a two-tap [3 1]/4 filter fed by their original value and the adjacent reference sample. In horizontal (Intra_Angular[10]) prediction, the boundary samples of the first column of the TB are modified such that half of the difference between their neighbored reference sample and the top-left reference sample is added. This makes the prediction signal more smooth when large variations in the vertical direction are present. In vertical (Intra_Angular[26]) prediction, the same is applied to the first row of samples.

6) *Reference Sample Substitution:* The neighboring reference samples are not available at the slice or tile boundaries. In addition, when a loss-resilience feature known as constrained intra prediction is enabled, the neighboring reference samples inside any interpicture-predicted PB are also considered not available in order to avoid letting potentially corrupted prior decoded picture data propagate errors into the prediction signal. While only Intra_DC prediction mode is allowed for such cases in H.264/MPEG-4 AVC, HEVC allows the use of other intrapicture prediction modes after substituting the nonavailable reference sample values with the neighboring available reference sample values.

7) *Mode Coding:* HEVC supports a total of 33 Intra_Angular prediction modes and Intra_Planar and Intra_DC prediction modes for luma prediction for all block sizes. Due to the increased number of directions, HEVC considers three most probable modes (MPMs) when coding the luma intrapicture prediction mode predictively, rather than the one most probable mode considered in H.264/MPEG-4 AVC.

Among the three most probable modes, the first two are initialized by the luma intrapicture prediction modes of the above and left PBs if those PBs are available and are coded using an intrapicture prediction mode. Any unavailable prediction mode is considered to be Intra_DC. The PB above the luma CTB is always considered to be unavailable in order to avoid the need to store a line buffer of neighboring luma prediction modes.

When the first two most probable modes are not equal, the third most probable mode is set equal to Intra_Planar, Intra_DC, or Intra_Angular[26] (vertical), according to which of these modes, in this order, is not a duplicate of one of the first two modes. When the first two most probable modes are the same, if this first mode has the value Intra_Planar or Intra_DC, the second and third most probable modes are assigned as Intra_Planar, Intra_DC, or Intra_Angular[26], according to which of these modes, in this order, are not duplicates. When the first two most probable modes are the same and the first mode has an Intra_Angular value, the second and third most probable modes are chosen as the two angular prediction modes that are closest to the angle (i.e., the value of k) of the first.

In the case that the current luma prediction mode is one of three MPMs, only the MPM index is transmitted to the decoder. Otherwise, the index of the current luma prediction mode excluding the three MPMs is transmitted to the decoder by using a 5-b fixed length code.

For chroma intrapicture prediction, HEVC allows the encoder to select one of five modes: Intra_Planar, Intra_Angular[26] (vertical), Intra_Angular[10] (horizontal), Intra_DC, and Intra_Derived. The Intra_Derived mode specifies that the chroma prediction uses the same angular direction as the luma prediction. With this scheme, all angular modes specified for luma in HEVC can, in principle, also be used in the chroma prediction, and a good tradeoff is achieved between prediction accuracy and the signaling overhead. The selected chroma prediction mode is coded directly (without using an MPM prediction mechanism).

H. Interpicture Prediction

1) *PB Partitioning:* Compared to intrapicture-predicted CBs, HEVC supports more PB partition shapes for interpicture-predicted CBs. The partitioning modes of PART_2N×2N, PART_2N×N, and PART_N×2N indicate the cases when the CB is not split, split into two equal-size PBs horizontally, and split into two equal-size PBs vertically, respectively. PART_N×N specifies that the CB is split into four equal-size PBs, but this mode is only supported when the CB size is equal to the smallest allowed CB size. In addition, there are four partitioning types that support splitting the CB into two PBs having different sizes: PART_2N×nU, PART_2N×nD, PART_nL×2N, and PART_nR×2N. These types are known as asymmetric motion partitions.

2) *Fractional Sample Interpolation:* The samples of the PB for an intrapicture-predicted CB are obtained from those of a

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000133

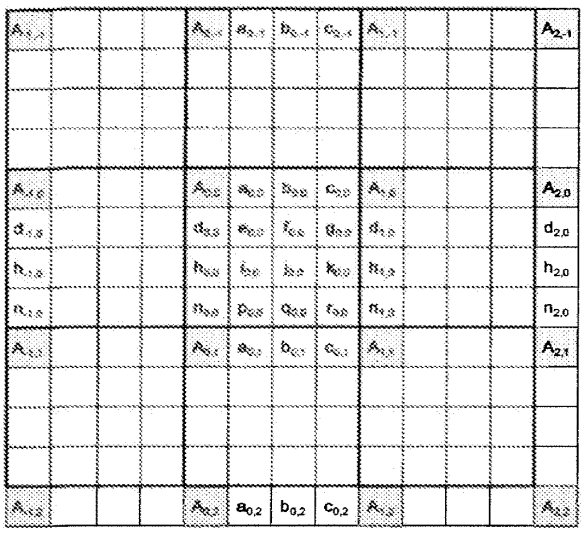


Fig. 7. Integer and fractional sample positions for luma interpolation.

corresponding block region in the reference picture identified by a reference picture index, which is at a position displaced by the horizontal and vertical components of the motion vector. Except for the case when the motion vector has an integer value, fractional sample interpolation is used to generate the prediction samples for noninteger sampling positions. As in H.264/MPEG-4 AVC, HEVC supports motion vectors with units of one quarter of the distance between luma samples. For chroma samples, the motion vector accuracy is determined according to the chroma sampling format, which for 4:2:0 sampling results in units of one eighth of the distance between chroma samples.

The fractional sample interpolation for luma samples in HEVC uses separable application of an eight-tap filter for the half-sample positions and a seven-tap filter for the quarter-sample positions. This is in contrast to the process used in H.264/MPEG-4 AVC, which applies a two-stage interpolation process by first generating the values of one or two neighboring samples at half-sample positions using six-tap filtering, rounding the intermediate results, and then averaging two values at integer or half-sample positions. HEVC instead uses a single consistent separable interpolation process for generating all fractional positions without intermediate rounding operations, which improves precision and simplifies the architecture of the fractional sample interpolation. The interpolation precision is also improved in HEVC by using longer filters, i.e., seven-tap or eight-tap filtering rather than the six-tap filtering used in H.264/MPEG-4 AVC. Using only seven taps rather than the eight used for half-sample positions was sufficient for the quarter-sample interpolation positions since the quarter-sample positions are relatively close to integer-sample positions, so the most distant sample in an eight-tap interpolator would effectively be farther away than in the half-sample case (where the relative distances of the integer-sample positions are symmetric). The actual filter tap values of the interpolation filtering kernel were partially derived from DCT basis function equations.

TABLE II
FILTER COEFFICIENTS FOR LUMA FRACTIONAL SAMPLE INTERPOLATION

Index i	−3	−2	−1	0	1	2	3	4
hfilter[i]	−1	4	−11	40	40	−11	4	1
qfilter[i]	−1	4	−10	58	17	−5	1	

In Fig. 7, the positions labeled with upper-case letters, $A_{i,j}$, represent the available luma samples at integer sample locations, whereas the other positions labeled with lower-case letters represent samples at noninteger sample locations, which need to be generated by interpolation.

The samples labeled $a_{0,j}$, $b_{0,j}$, $c_{0,j}$, $d_{0,0}$, $h_{0,0}$, and $n_{0,0}$ are derived from the samples $A_{i,j}$ by applying the eight-tap filter for half-sample positions and the seven-tap filter for the quarter-sample positions as follows:

$$a_{0,j} = (\textstyle\sum_{i=-3..3} A_{i,j}\, \text{qfilter}[i]) >> (B-8)$$
$$b_{0,j} = (\textstyle\sum_{i=-3..4} A_{i,j}\, \text{hfilter}[i]) >> (B-8)$$
$$c_{0,j} = (\textstyle\sum_{i=-2..4} A_{i,j}\, \text{qfilter}[1-i]) >> (B-8)$$
$$d_{0,0} = (\textstyle\sum_{i=-3..3} A_{0,j}\, \text{qfilter}[j]) >> (B-8)$$
$$h_{0,0} = (\textstyle\sum_{i=-3..4} A_{0,j}\, \text{hfilter}[j]) >> (B-8)$$
$$n_{0,0} = (\textstyle\sum_{j=-2..4} A_{0,j}\, \text{qfilter}[1-j]) >> (B-8)$$

where the constant $B \geq 8$ is the bit depth of the reference samples (and typically $B = 8$ for most applications) and the filter coefficient values are given in Table II. In these formulas, $>>$ denotes an arithmetic right shift operation.

The samples labeled $e_{0,0}$, $f_{0,0}$, $g_{0,0}$, $i_{0,0}$, $j_{0,0}$, $k_{0,0}$, $p_{0,0}$, $q_{0,0}$, and $r_{0,0}$ can be derived by applying the corresponding filters to samples located at vertically adjacent $a_{0,j}$, $b_{0,j}$ and $c_{0,j}$ positions as follows:

$$e_{0,0} = (\textstyle\sum_{v=-3..3} a_{0,v}\, \text{qfilter}[v]) >> 6$$
$$f_{0,0} = (\textstyle\sum_{v=-3..3} b_{0,v}\, \text{qfilter}[v]) >> 6$$
$$g_{0,0} = (\textstyle\sum_{v=-3..3} c_{0,v}\, \text{qfilter}[v]) >> 6$$
$$i_{0,0} = (\textstyle\sum_{v=-3..4} a_{0,v}\, \text{hfilter}[v]) >> 6$$
$$j_{0,0} = (\textstyle\sum_{v=-3..4} b_{0,v}\, \text{hfilter}[v]) >> 6$$
$$k_{0,0} = (\textstyle\sum_{v=-3..4} c_{0,v}\, \text{hfilter}[v]) >> 6$$
$$p_{0,0} = (\textstyle\sum_{v=-2..4} a_{0,v}\, \text{qfilter}[1-v]) >> 6$$
$$q_{0,0} = (\textstyle\sum_{v=-2..4} b_{0,v}\, \text{qfilter}[1-v]) >> 6$$
$$r_{0,0} = (\textstyle\sum_{v=-2..4} c_{0,v}\, \text{qfilter}[1-v]) >> 6.$$

The interpolation filtering is separable when B is equal to 8, so the same values could be computed in this case by applying the vertical filtering before the horizontal filtering. When implemented appropriately, the motion compensation process of HEVC can be performed using only 16-b storage elements (although care must be taken to do this correctly).

It is at this point in the process that weighted prediction is applied when selected by the encoder. Whereas H.264/MPEG-4 AVC supported both temporally implicit and explicit weighted prediction, in HEVC only explicit weighted prediction is applied, by scaling and offsetting the prediction with values sent explicitly by the encoder. The bit depth of the prediction is then adjusted to the original bit depth of the reference samples. In the case of uniprediction, the interpolated (and possibly weighted) prediction value is rounded,

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000134

TABLE III
FILTER COEFFICIENTS FOR CHROMA FRACTIONAL SAMPLE INTERPOLATION

Index	−1	0	1	2
filter1[i]	−2	58	10	−2
filter2[i]	−4	54	16	−2
filter3[i]	−6	46	28	−4
filter4[i]	−4	36	36	−4

right-shifted, and clipped to have the original bit depth. In the case of bipredictiion, the interpolated (and possibly weighted) prediction values from two PBs are added first, and then rounded, right-shifted, and clipped.

In H.264/MPEG-4 AVC, up to three stages of rounding operations are required to obtain each prediction sample (for samples located at quarter-sample positions). If biprediction is used, the total number of rounding operations is then seven in the worst case. In HEVC, at most two rounding operations are needed to obtain each sample located at the quarter-sample positions, thus five rounding operations are sufficient in the worst case when biprediction is used. Moreover, in the most common usage, where the bit depth B is 8 b, the total number of rounding operations in the worst case is further reduced to 3. Due to the lower number of rounding operations, the accumulated rounding error is decreased and greater flexibility is enabled in regard to the manner of performing the necessary operations in the decoder.

The fractional sample interpolation process for the chroma components is similar to the one for the luma component, except that the number of filter taps is 4 and the fractional accuracy is 1/8 for the usual 4:2:0 chroma format case. HEVC defines a set of four-tap filters for eighth-sample positions, as given in Table III for the case of 4:2:0 chroma format (where, in H.264/MPEG-4 AVC, only two-tap bilinear filtering was applied).

Filter coefficient values denoted as filter1[i], filter2[i], filter3[i], and filter4[i] with $i = -1,\ldots, 2$ are used for interpolating the 1/8th, 2/8th, 3/8th, and 4/8th fractional positions for the chroma samples, respectively. Using symmetry for the 5/8th, 6/8th, and 7/8th fractional positions, the mirrored values of filter3[$1-i$], filter2[$1-i$], and filter1[$1-i$] with $i = -1, \ldots, 2$ are used, respectively.

3) *Merge Mode:* Motion information typically consists of the horizontal and vertical motion vector displacement values, one or two reference picture indices, and, in the case of prediction regions in B slices, an identification of which reference picture list is associated with each index. HEVC includes a merge mode to derive the motion information from spatially or temporally neighboring blocks. It is denoted as merge mode since it forms a merged region sharing all motion information.

The merge mode is conceptually similar to the direct and skip modes in H.264/MPEG-4 AVC. However, there are two important differences. First, it transmits index information to select one out of several available candidates, in a manner sometimes referred to as a motion vector competition scheme. It also explicitly identifies the reference picture list and reference picture index, whereas the direct mode assumes that these have some predefined values.

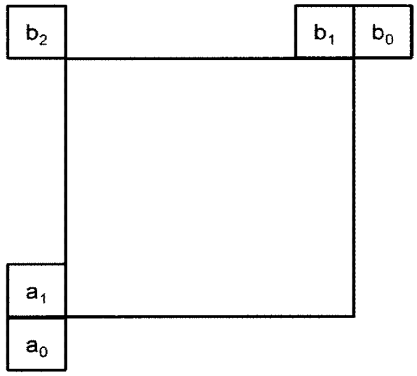


Fig. 8. Positions of spatial candidates of motion information.

The set of possible candidates in the merge mode consists of spatial neighbor candidates, a temporal candidate, and generated candidates. Fig. 8 shows the positions of five spatial candidates. For each candidate position, the availability is checked according to the order $\{a_1, b_1, b_0, a_0, b_2\}$. If the block located at the position is intrapicture predicted or the position is outside of the current slice or tile, it is considered as unavailable.

After validating the spatial candidates, two kinds of redundancy are removed. If the candidate position for the current PU would refer to the first PU within the same CU, the position is excluded, as the same merge could be achieved by a CU without splitting into prediction partitions. Furthermore, any redundant entries where candidates have exactly the same motion information are also excluded.

For the temporal candidate, the right bottom position just outside of the collocated PU of the reference picture is used if it is available. Otherwise, the center position is used instead. The way to choose the collocated PU is similar to that of prior standards, but HEVC allows more flexibility by transmitting an index to specify which reference picture list is used for the collocated reference picture.

One issue related to the use of the temporal candidate is the amount of the memory to store the motion information of the reference picture. This is addressed by restricting the granularity for storing the temporal motion candidates to only the resolution of a 16×16 luma grid, even when smaller PB structures are used at the corresponding location in the reference picture. In addition, a PPS-level flag allows the encoder to disable the use of the temporal candidate, which is useful for applications with error-prone transmission.

The maximum number of merge candidates C is specified in the slice header. If the number of merge candidates found (including the temporal candidate) is larger than C, only the first $C - 1$ spatial candidates and the temporal candidate are retained. Otherwise, if the number of merge candidates identified is less than C, additional candidates are generated until the number is equal to C. This simplifies the parsing and makes it more robust, as the ability to parse the coded data is not dependent on merge candidate availability.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000135

For B slices, additional merge candidates are generated by choosing two existing candidates according to a predefined order for reference picture list 0 and list 1. For example, the first generated candidate uses the first merge candidate for list 0 and the second merge candidate for list 1. HEVC specifies a total of 12 predefined pairs of two in the following order in the already constructed merge candidate list as (0, 1), (1, 0), (0, 2), (2, 0), (1, 2), (2, 1), (0, 3), (3, 0), (1, 3), (3, 1), (2, 3), and (3, 2). Among them, up to five candidates can be included after removing redundant entries.

When the slice is a P slice or the number of merge candidates is still less than C, zero motion vectors associated with reference indices from zero to the number of reference pictures minus one are used to fill any remaining entries in the merge candidate list.

In HEVC, the skip mode is treated as a special case of the merge mode when all coded block flags are equal to zero. In this specific case, only a skip flag and the corresponding merge index are transmitted to the decoder. The B-direct mode of H.264/MPEG-4 AVC is also replaced by the merge mode, since the merge mode allows all motion information to be derived from the spatial and temporal motion information of the neighboring blocks with residual coding.

4) *Motion Vector Prediction for Nonmerge Mode:* When an interpicture-predicted CB is not coded in the skip or merge modes, the motion vector is differentially coded using a motion vector predictor. Similar to the merge mode, HEVC allows the encoder to choose the motion vector predictor among multiple predictor candidates. The difference between the predictor and the actual motion vector and the index of the candidate are transmitted to the decoder.

Only two spatial motion candidates are chosen according to the availability among five candidates in Fig. 8. The first spatial motion candidate is chosen from the set of left positions $\{a_0, a_1\}$ and the second one from the set of above positions $\{b_0, b_1, b_2\}$ according to their availabilities, while keeping the searching order as indicated in the two sets.

HEVC only allows a much lower number of candidates to be used in the motion vector prediction process for the nonmerge case, since the encoder can send a coded difference to change the motion vector. Furthermore, the encoder needs to perform motion estimation, which is one of the most computationally expensive operations in the encoder, and complexity is reduced by allowing a small number of candidates.

When the reference index of the neighboring PU is not equal to that of the current PU, a scaled version of the motion vector is used. The neighboring motion vector is scaled according to the temporal distances between the current picture and the reference pictures indicated by the reference indices of the neighboring PU and the current PU, respectively. When two spatial candidates have the same motion vector components, one redundant spatial candidate is excluded.

When the number of motion vector predictors is not equal to two and the use of temporal MV prediction is not explicitly disabled, the temporal MV prediction candidate is included. This means that the temporal candidate is not used at all when two spatial candidates are available. Finally, a zero motion vector is included repeatedly until the number of motion vector prediction candidates is equal to two, which guarantees that the number of motion vector predictors is two. Thus, only a coded flag is necessary to identify which motion vector prediction is used in the case of nonmerge mode.

I. Transform, Scaling, and Quantization

HEVC uses transform coding of the prediction error residual in a similar manner as in prior standards. The residual block is partitioned into multiple square TBs, as described in Section IV-E. The supported transform block sizes are 4×4, 8×8, 16×16, and 32×32.

1) *Core Transform:* Two-dimensional transforms are computed by applying 1-D transforms in the horizontal and vertical directions. The elements of the core transform matrices were derived by approximating scaled DCT basis functions, under considerations such as limiting the necessary dynamic range for transform computation and maximizing the precision and closeness to orthogonality when the matrix entries are specified as integer values.

$$H = \begin{bmatrix}
64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\
90 & 87 & 80 & 70 & 57 & 43 & 25 & 9 & -9 & -25 & -43 & -57 & -70 & -80 & -87 & 90 \\
89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 & -89 & -75 & -50 & -18 & 18 & 50 & 75 & 89 \\
87 & 57 & 9 & -43 & -80 & -90 & -70 & -25 & 25 & 70 & 90 & 80 & 43 & -9 & -57 & -87 \\
83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 & 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\
80 & 9 & -70 & -87 & -25 & 57 & 90 & 43 & -43 & -90 & -57 & 25 & 87 & 70 & -9 & -80 \\
75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 & -75 & 18 & 89 & 50 & -50 & -89 & -18 & 75 \\
70 & -43 & -87 & 9 & 90 & 25 & -80 & -57 & 57 & 80 & -25 & -90 & -9 & 87 & 43 & -70 \\
64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\
57 & -80 & -25 & 90 & -9 & -87 & 43 & 70 & -70 & -43 & 87 & 9 & -90 & 25 & 80 & -57 \\
50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 & -50 & 89 & -18 & -75 & 75 & 18 & -89 & 50 \\
43 & -90 & 57 & 25 & -87 & 70 & 9 & -80 & 80 & -9 & -70 & 87 & -25 & -57 & 90 & -43 \\
36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 & 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\
25 & -70 & 90 & -80 & 43 & 9 & -57 & 87 & -87 & 57 & -9 & -43 & 80 & -90 & 70 & -25 \\
18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 & -18 & 50 & -75 & 89 & -89 & 75 & -50 & 18 \\
9 & -25 & 43 & -57 & 70 & -80 & 87 & -90 & 90 & -87 & 80 & -70 & 57 & -43 & 25 & -9
\end{bmatrix}.$$

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000136

For simplicity, only one integer matrix for the length of 32 points is specified, and subsampled versions are used for other sizes. For example, the matrix for the length-16 transform is as shown in the equation at the bottom of the previous page.

The matrices for the length-8 and length-4 transforms can be derived by using the first eight entries of rows 0, 2, 4, ..., and using the first four entries of rows 0, 4, 8, ..., respectively. Although the standard specifies the transform simply in terms of the value of a matrix, the values of the entries in the matrix were selected to have key symmetry properties that enable fast partially factored implementations with far fewer mathematical operations than an ordinary matrix multiplication, and the larger transforms can be constructed by using the smaller transforms as building blocks.

Due to the increased size of the supported transforms, limiting the dynamic range of the intermediate results from the first stage of the transformation is quite important. HEVC explicitly inserts a 7-b right shift and 16-b clipping operation after the first 1-D inverse transform stage of the transform (the vertical inverse transform stage) to ensure that all intermediate values can be stored in 16-b memory (for 8-b video decoding).

2) *Alternative 4 × 4 Transform:* For the transform block size of 4×4, an alternative integer transform derived from a DST is applied to the luma residual blocks for intrapicture prediction modes, with the transform matrix

$$H = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}.$$

The basis functions of the DST better fit the statistical property that the residual amplitudes tend to increase as the distance from the boundary samples that are used for prediction becomes larger. In terms of complexity, the 4×4 DST-style transform is not much more computationally demanding than the 4×4 DCT-style transform, and it provides approximately 1% bit-rate reduction in intrapicture predictive coding.

The usage of the DST type of transform is restricted to only 4×4 luma transform blocks, since for other cases the additional coding efficiency improvement for including the additional transform type was found to be marginal.

3) *Scaling and Quantization:* Since the rows of the transform matrix are close approximations of values of uniformly scaled basis functions of the orthonormal DCT, the prescaling operation that is incorporated in the dequantization of H.264/MPEG-4 AVC is not needed in HEVC. This avoidance of frequency-specific basis function scaling is useful in reducing the intermediate memory size, especially when considering that the size of the transform can be as large as 32×32.

For quantization, HEVC uses essentially the same URQ scheme controlled by a quantization parameter (QP) as in H.264/MPEG-4 AVC. The range of the QP values is defined from 0 to 51, and an increase by 6 doubles the quantization step size such that the mapping of QP values to step sizes is approximately logarithmic. Quantization scaling matrices are also supported.

To reduce the memory needed to store frequency-specific scaling values, only quantization matrices of sizes 4×4 and 8×8 are used. For the larger transformations of 16×16 and 32×32 sizes, an 8×8 scaling matrix is sent and is applied by sharing values within 2×2 and 4×4 coefficient groups in frequency subspaces—except for values at DC (zero-frequency) positions, for which distinct values are sent and applied.

J. Entropy Coding

HEVC specifies only one entropy coding method, CABAC [13] rather than two as in H.264/MPEG-4 AVC. The core algorithm of CABAC is unchanged, and the following subsections present several aspects of how it is used in the HEVC design.

1) *Context Modeling:* Appropriate selection of context modeling is known to be a key factor to improve the efficiency of CABAC coding. In HEVC, the splitting depth of the coding tree or transform tree is exploited to derive the context model indices of various syntax elements in addition to the spatially neighboring ones used in H.264/AVC. For example, the syntax element *skip_flag* specifying whether the CB is coded as interpicture predictively skipped and the syntax element *split_coding_unit_flag* specifying whether the CB is further split are coded by using context models based on the spatially neighboring information. The syntax element *split_transform_flag* specifying whether the TB is further split and three syntax elements specifying non-zero transform coefficients for each color component, *cbf_luma*, *cbf_cb* and *cbf_cr*, are coded based on the splitting depth of the transform tree. Although the number of contexts used in HEVC is substantially less than in H.264/MPEG-4 AVC, the entropy coding design actually provides better compression than would a straightforward extension of the H.264/MPEG-4 AVC scheme. Moreover, more extensive use is made in HEVC of the bypass mode of CABAC operation to increase throughput by reducing the amount of data that needs to be coded using CABAC contexts. Dependences between coded data are also carefully considered to enable further throughput maximization.

2) *Adaptive Coefficient Scanning:* Coefficient scanning is performed in 4×4 subblocks for all TB sizes (i.e., using only one coefficient region for the 4×4 TB size, and using multiple 4×4 coefficient regions within larger transform blocks). Three coefficient scanning methods, diagonal up-right, horizontal, and vertical scans as shown in Fig. 9, are selected implicitly for coding the transform coefficients of 4×4 and 8×8 TB sizes in intrapicture-predicted regions. The selection of the coefficient scanning order depends on the directionalities of the intrapicture prediction. The vertical scan is used when the prediction direction is close to horizontal and the horizontal scan is used when the prediction direction is close to vertical. For other prediction directions, the diagonal up-right scan is used.

For the transform coefficients in interpicture prediction modes of all block sizes and for the transform coefficients of 16×16 or 32×32 intrapicture prediction, the 4×4 diagonal up-right scan is exclusively applied to subblocks of transform coefficients.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000137

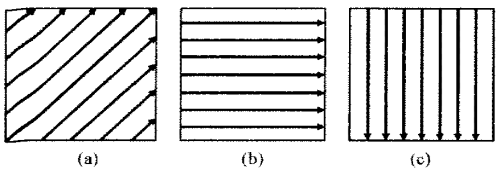


Fig. 9. Three coefficient scanning methods in HEVC. (a) Diagonal up-right scan. (b) Horizontal scan. (c) Vertical scan.

3) *Coefficient Coding:* Similar to H.264/MPEG-4 AVC, HEVC transmits the position of the last nonzero transform coefficient, a significance map, sign bits and levels for the transform coefficients. However, various changes for each part have been made, especially for better handling of the significantly increased size of the TBs.

First, the horizontal and vertical frequency coordinate positions of the last nonzero coefficient are coded for the TB before sending the significance maps of 4×4 subblocks that indicate which other transform coefficients have nonzero values, rather than sending a series of last-coefficient identification flags that are interleaved with the significance map as done in H.264/MPEG-4 AVC.

The significance map is derived for significance groups relating to the fixed size 4×4 subblocks. For all groups having at least one coefficient preceding the last coefficient position, a significant group flag specifying a nonzero coefficient group is transmitted, followed by coefficient significance flags for each coefficient prior to the indicated position of the last significant coefficient. The context models for the significant coefficient flags are dependent on the coefficient position as well as the values of the right and the bottom significant group flags.

A method known as sign data hiding is used for further compression improvement. The sign bits are coded conditionally based on the number and positions of coded coefficients. When sign data hiding is used and there are at least two nonzero coefficients in a 4×4 subblock and the difference between the scan positions of the first and the last nonzero coefficients is greater than 3, the sign bit of the first nonzero coefficient is inferred from the parity of the sum of the coefficient amplitudes. Otherwise, the sign bit is coded normally. At the encoder side, this can be implemented by selecting one coefficient with an amplitude close to the boundary of a quantization interval to be forced to use the adjacent quantization interval in cases where the parity would not otherwise indicate the correct sign of the first coefficient. This allows the sign bit to be encoded at a lower cost (in rate-distortion terms) than if it were coded separately—by giving the encoder the freedom to choose which transform coefficient amplitude can be altered with the lowest rate-distortion cost.

For each position where the corresponding significant coefficient flag is equal to one, two flags specifying whether the level value is greater than one or two are coded, and then the remaining level value is coded depending on those two values.

K. In-Loop Filters

In HEVC, two processing steps, namely a deblocking filter (DBF) followed by an SAO filter, are applied to the reconstructed samples before writing them into the decoded picture buffer in the decoder loop. The DBF is intended to reduce the blocking artifacts due to block-based coding. The DBF is similar to the DBF of the H.264/MPEG-4 AVC standard, whereas SAO is newly introduced in HEVC. While the DBF is only applied to the samples located at block boundaries, the SAO filter is applied adaptively to all samples satisfying certain conditions, e.g., based on gradient. During the development of HEVC, it had also been considered to operate a third processing step called the adaptive loop filter (ALF) after the SAO filter; however, the ALF feature was not included in the final design.

1) *Deblocking Filter:* The deblocking filter is applied to all samples adjacent to a PU or TU boundary except the case when the boundary is also a picture boundary, or when deblocking is disabled across slice or tile boundaries (which is an option that can be signaled by the encoder). It should be noted that both PU and TU boundaries should be considered since PU boundaries are not always aligned with TU boundaries in some cases of interpicture-predicted CBs. Syntax elements in the SPS and slice headers control whether the deblocking filter is applied across the slice and tile boundaries.

Unlike H.264/MPEG-4 AVC, where the deblocking filter is applied on a 4×4 sample grid basis, HEVC only applies the deblocking filter to the edges that are aligned on an 8×8 sample grid, for both the luma and chroma samples. This restriction reduces the worst-case computational complexity without noticeable degradation of the visual quality. It also improves parallel-processing operation by preventing cascading interactions between nearby filtering operations.

The strength of the deblocking filter is controlled by the values of several syntax elements similar to the scheme in H.264/MPEG-4 AVC, but only three strengths are used rather than five. Given that P and Q are two adjacent blocks with a common 8×8 grid boundary, the filter strength of 2 is assigned when one of the blocks is intrapicture predicted. Otherwise, the filter strength of 1 is assigned if any of the following conditions is satisfied.

1) P or Q has at least one nonzero transform coefficient.
2) The reference indices of P and Q are not equal.
3) The motion vectors of P and Q are not equal.
4) The difference between a motion vector component of P and Q is greater than or equal to one integer sample.

If none of the above conditions is met, the filter strength of 0 is assigned, which means that the deblocking process is not applied.

According to the filter strength and the average quantization parameter of P and Q, two thresholds, t_C and β, are determined from predefined tables. For luma samples, one of three cases, no filtering, strong filtering, and weak filtering, is chosen based on β. Note that this decision is shared across four luma rows or columns using the first and the last rows or columns to reduce the computational complexity.

There are only two cases, no filtering and normal filtering, for chroma samples. Normal filtering is applied only when the filter strength is greater than one. The filtering process is then performed using the control variables t_C and β.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000138

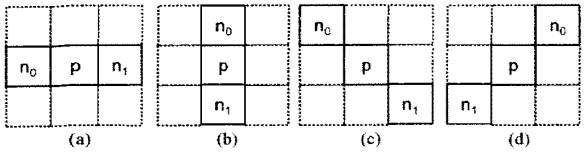


Fig. 10. Four gradient patterns used in SAO. Sample labeled "p" indicates a center sample to be considered. Two samples labeled "n_0" and "n_1" specify two neighboring samples along the (a) horizontal (sao_eo_class = 0), (b) vertical (sao_eo_class = 1), (c) 135° diagonal (sao_eo_class = 2), and (d) 45° (sao_eo_class = 3) gradient patterns.

In HEVC, the processing order of the deblocking filter is defined as horizontal filtering for vertical edges for the entire picture first, followed by vertical filtering for horizontal edges. This specific order enables either multiple horizontal filtering or vertical filtering processes to be applied in parallel threads, or can still be implemented on a CTB-by-CTB basis with only a small processing latency.

2) *SAO:* SAO is a process that modifies the decoded samples by conditionally adding an offset value to each sample after the application of the deblocking filter, based on values in look-up tables transmitted by the encoder. SAO filtering is performed on a region basis, based on a filtering type selected per CTB by a syntax element sao_type_idx. A value of 0 for sao_type_idx indicates that the SAO filter is not applied to the CTB, and the values 1 and 2 signal the use of the band offset and edge offset filtering types, respectively.

In the band offset mode specified by sao_type_idx equal to 1, the selected offset value directly depends on the sample amplitude. In this mode, the full sample amplitude range is uniformly split into 32 segments called bands, and the sample values belonging to four of these bands (which are consecutive within the 32 bands) are modified by adding transmitted values denoted as band offsets, which can be positive or negative. The main reason for using four consecutive bands is that in the smooth areas where banding artifacts can appear, the sample amplitudes in a CTB tend to be concentrated in only few of the bands. In addition, the design choice of using four offsets is unified with the edge offset mode of operation which also uses four offset values.

In the edge offset mode specified by sao_type_idx equal to 2, a syntax element sao_eo_class with values from 0 to 3 signals whether a horizontal, vertical or one of two diagonal gradient directions is used for the edge offset classification in the CTB. Fig. 10 depicts the four gradient patterns used for the respective sao_eo_class in this mode. Each sample in the CTB is classified into one of five EdgeIdx categories by comparing the sample value p located at some position with the values n0 and n1 of two samples located at neighboring positions as shown in Table IV. This classification is done for each sample based on decoded sample values, so no additional signaling is required for the EdgeIdx classification. Depending on the EdgeIdx category at the sample position, for EdgeIdx categories from 1 to 4, an offset value from a transmitted look-up table is added to the sample value. The offset values are always positive for categories 1 and 2 and negative for categories 3 and 4 - thus the filter generally has a smoothing effect in the edge offset mode.

TABLE IV
SAMPLE EDGEIDX CATEGORIES IN SAO EDGE CLASSES

EdgeIdx	Condition	Meaning
0	Cases not listed below	Monotonic area
1	$p < n_0$ and $p < n_1$	Local min
2	$p < n_0$ and $p = n_1$ or $p < n_1$ and $p = n_0$	Edge
3	$p > n_0$ and $p = n_1$ or $p > n_1$ and $p = n_0$	Edge
4	$p > n_0$ and $p > n_1$	Local max

Thus, for SAO types 1 and 2, a total of four amplitude offset values are transmitted to the decoder for each CTB. For type 1, the sign is also encoded. The offset values and related syntax elements such as sao_type_idx and sao_eo_class are determined by the encoder - typically using criteria that optimize rate-distortion performance. The SAO parameters can be indicated to be inherited from the left or above CTB using a merge flag to make the signaling efficient. In summary, SAO is a nonlinear filtering operation which allows additional refinement of the reconstructed signal, and it can enhance the signal representation in both smooth areas and around edges.

L. Special Coding Modes

HEVC defines three special coding modes, which can be invoked at the CU level or the TU level.

1) In I_PCM mode, the prediction, transform, quantization and entropy coding are bypassed, and the samples are directly represented by a pre-defined number of bits. Its main purpose is to avoid excessive consumption of bits when the signal characteristics are extremely unusual and cannot be properly handled by hybrid coding (e.g., noise-like signals).
2) In lossless mode, the transform, quantization, and other processing that affects the decoded picture (SAO and deblocking filters) are bypassed, and the residual signal from inter- or intrapicture prediction is directly fed into the entropy coder (using the same neighborhood contexts that would usually be applied to the quantized transform coefficients). This allows mathematically lossless reconstruction, which is achieved without defining any additional coding tools.
3) In transform skipping mode, only the transform is bypassed. This primarily improves compression for certain types of video content such as computer-generated images or graphics mixed with camera-view content (e.g., scrolling text). This mode can be applied to TBs of 4×4 size only.

SAO and deblocking filtering are not applied to lossless mode regions, and a flag controls whether they are applied to I_PCM regions.

V. PROFILES, TIERS, AND LEVELS

A. Profile, Level, and Tier Concepts

Profiles, tiers, and levels specify conformance points for implementing the standard in an interoperable way across various applications that have similar functional requirements. A profile defines a set of coding tools or algorithms that can be used in generating a conforming bitstream, whereas a level places

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000139

constraints on certain key parameters of the bitstream, corresponding to decoder processing load and memory capabilities. Level restrictions are established in terms of maximum sample rate, maximum picture size, maximum bit rate, minimum compression ratio and capacities of the DPB, and the coded picture buffer (CPB) that holds compressed data prior to its decoding for data flow management purposes. In the design of HEVC, it was determined that some applications existed that had requirements that differed only in terms of maximum bit rate and CPB capacities. To resolve this issue, two tiers were specified for some levels—a Main Tier for most applications and a High Tier for use in the most demanding applications.

A decoder conforming to a certain tier and level is required to be capable of decoding all bitstreams that conform to the same tier or the lower tier of that level or any level below it. Decoders conforming to a specific profile must support all features in that profile. Encoders are not required to make use of any particular set of features supported in a profile, but are required to produce conforming bitstreams, i.e., bitstreams that obey the specified constraints that enable them to be decoded by conforming decoders.

B. The HEVC Profile and Level Definitions

Only three profiles targetting different application requirements, called the Main, Main 10, and Main Still Picture profiles, are foreseen to be finalized by January 2013. Minimizing the number of profiles provides a maximum amount of interoperability between devices, and is further justified by the fact that traditionally separate services, such as broadcast, mobile, streaming, are converging to the point where most devices should become usable to support all of them. The three drafted profiles consist of the coding tools and high-layer syntax described in the earlier sections of this paper, while imposing the following restrictions.

1) Only 4:2:0 chroma sampling is supported.
2) When an encoder encodes a picture using multiple tiles, it cannot also use wavefront parallel processing, and each tile must be at least 256 luma samples wide and 64 luma samples tall.
3) In the Main and Main Still Picture profiles, only a video precision of 8 b per sample is supported, while the Main 10 profile supports up to 10 b per sample.
4) In the Main Still Picture profile, the entire bitstream must contain only one coded picture (and thus interpicture prediction is not supported).

Currently, the definition of 13 levels is planned to be included in the first version of the standard as shown in Table V, ranging from levels that support only relatively small picture sizes such as a luma picture size of 176×144 (sometimes called a quarter common intermediate format) to picture sizes as large as 7680×4320 (often called 8k×4k). The picture width and height are each required to be less than or equal to $\sqrt{8 \cdot \text{MaxLumaPS}}$, where MaxLumaPS is the maximum luma picture size as shown in Table V (to avoid the problems for decoders that could be involved with extreme picture shapes).

There are two tiers supported for eight of these levels (level 4 and higher). The CPB capacity is equal to the maximum bit rate times 1 s for all levels except level 1, which has a (higher) CPB capacity of 350 000 b. The specified maximum DPB capacity in each level is six pictures when operating at the maximum picture size supported by the level (including both the current picture and all other pictures that are retained in the decoder at any point in time for reference or output purposes). When operating with a smaller picture size than the maximum size supported by the level, the DPB picture storage capacity can increase to as many as 16 pictures (depending on the particular selected picture size). Level-specific constraints are also specified for the maximum number of tiles used horizontally and vertically within each picture and the maximum number of tiles used per second.

TABLE V
LEVEL LIMITS FOR THE MAIN PROFILE

Level	Max Luma Picture Size (samples)	Max Luma Sample Rate (samples/s)	Main Tier Max Bit Rate (1000 bits/s)	High Tier Max Bit Rate (1000 bits/s)	Min Comp. Ratio
1	36 864	552 960	128	–	2
2	122 880	3 686 400	1500	–	2
2.1	245 760	7 372 800	3000	–	2
3	552 960	16 588 800	6000	–	2
3.1	983 040	33 177 600	10 000	–	2
4	2 228 224	66 846 720	12 000	30 000	4
4.1	2 228 224	133 693 440	20 000	50 000	4
5	8 912 896	267 386 880	25 000	100 000	6
5.1	8 912 896	534 773 760	40 000	160 000	8
5.2	8 912 896	1 069 547 520	60 000	240 000	8
6	35 651 584	1 069 547 520	60 000	240 000	8
6.1	35 651 584	2 139 095 040	120 000	480 000	8
6.2	35 651 584	4 278 190 080	240 000	800 000	6

VI. HISTORY AND STANDARDIZATION PROCESS

After the finalization of the H.264/MPEG-4 AVC High Profile in mid-2004, both ITU-T VCEG and ISO/IEC MPEG have been trying to identify when the next major advances in coding efficiency would become ready for standardization. VCEG began studying potential advances in 2004, began identifying certain key technology areas (KTAs) for study in early 2005, and developed a common KTA software codebase for this paper [14]. Various technologies were proposed and verified using the KTA software codebase, which was developed from the H.264/MPEG-4 AVC reference software known as the joint model (JM).

From 2005 to 2008, MPEG began exploration activities toward significant coding efficiency improvements as well, organized several workshops, and issued a "call for evidence" [15] of such advances in April 2009. Expert viewing tests were conducted to evaluate submissions of responses to the call.

From their respective investigations, it was agreed that there were sufficient technologies with the potential to improve the coding efficiency significantly, compared to the existing video coding standards. The Joint Collaborative Team on Video Coding (JCT-VC) was planned to be established by both groups in January 2010, and a joint call for proposals (CfP) on video compression technology [16] was issued by the same

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000140

TABLE VI
STRUCTURE OF CODING TOOLS ASSOCIATED WITH HIGH EFFICIENCY AND LOW COMPLEXITY CONFIGURATIONS OF HM 1

Functionality	High Efficiency	Low Complexity
CTB/CU structure	Tree structure from 8×8 to 64×64	
PU structure	Square and symmetric shapes	
TU structure	Three-level tree structure	Two-level tree structure
Transform	Integer transforms from four to 32 points	
Intra prediction	33 angular modes with DC mode	
Luma interpolation	12-tap separable	Six-tap directional
Chroma interpolation	Bilinear	
MV prediction	Spatial CU merging/AMVP	
Entropy coding	CABAC	Event-based VLC
Deblocking filter	Enabled	
Adaptive loop filter	Enabled	Disabled

time to identify the initial technologies that would serve as a basis of future standardization activities.

At its first meeting in April 2010, the JCT-VC established the HEVC project name, studied the proposals submitted in response to the CfP, and established the first version of a test model under consideration (TMuC) [17], which was produced collectively from elements of several promising proposals. A corresponding software codebase was implemented after this meeting. The technology submitted in several of the key proposal contributions was previously discussed in a special section of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY [18].

Although the TMuC showed significant coding efficiency improvements compared to prior standards, it had several redundant coding tools in each functional block of the video compression system, primarily due to the fact that the TMuC was a collective design from various contributions. During the second JCT-VC meeting in July 2010, the process began of selecting the minimal necessary set of coding tools for each functional block by thoroughly testing each component of the TMuC.

Based on the reported results of the exhaustive component testing [20], an HEVC test model version 1 (HM 1) [21] and the corresponding HEVC working draft specification version 1 (WD 1) [22] were produced as outputs of the third JCT-VC meeting in October 2010. Compared to the prior TMuC design, HM 1 was simplified greatly by removing coding tools that showed only marginal benefits relative to their computational complexity.

In several subsequent studies, the coding tools of the HM were classified into two categories called the high efficiency and low complexity configurations. Two corresponding test scenarios for verifying future contributions in the JCT-VC were also established. Table VI summarizes the HM 1 coding tools for the high efficiency and low complexity configurations.

From the fourth to the eleventh JCT-VC meetings, not just coding efficiency improvements, but many other aspects including computational complexity reduction, unification of various coding tools, and parallel friendly design were investigated, and the HEVC design was updated accordingly, until the current status of draft standard, as described in this paper, was reached. In this context, it also turned out that the differentiation for low complexity and high efficiency was no longer necessary became possible to define the unified main profile. Table VII provides a summary of coding tools of the high efficiency configuration in HM 1 and the current specification of HEVC.

TABLE VII
SUMMARY OF CODING TOOLS OF HIGH EFFICIENCY CONFIGURATION IN HM 1 AND HEVC

Functionality	HM 1 High Efficiency	HEVC (draft 9)
CTU structure	Tree structure from 8×8 to 64×64	Tree structure from 8×8 to 64×64
PU structure	Square and symmetric	Square, symmetric and asymmetric (only square for intra)
TU structure	Tree structure of square TUs	Tree structure of square TUs
Core transform	Integer transforms from 4 to 32 points (full factorization)	Integer transforms from 4 to 32 points (partially factorable)
Alternative transform	n/a	Integer DST type for 4×4
Intra prediction	33 angular modes with DC mode	33 angular modes with planar and DC modes
Luma interpolation	12-tap separable	8-tap/7-tap separable
Chroma interpolation	Bilinear	4-tap separable
MV prediction	AMVP	AMVP
MC region merging	Spatial CU merging	PU merging
Entropy coding	CABAC	CABAC
Deblocking filter	Nonparallel	Parallel
Sample adaptive offset	n/a	Enabled
Adaptive loop filter	Multiple shapes	n/a
Dedicated tools for parallel processing	Slices	Slices, tiles, wavefronts, and dependent slice segments

At the eighth JCT-VC meeting in February 2012, the draft version 6 of HEVC standard was produced, which was subsequently balloted as the ISO/IEC Committee Draft of the HEVC standard. The tenth JCT-VC Meeting in July 2012 released the draft version 8 for a Draft International Standard ballot, and the finalized text for Consent in ITU-T and Final Draft International Standard in ISO/IEC is expected to be produced in January 2013.

Future extensions of HEVC, which are already being explored and prepared by the JCT-VC's parent bodies, are likely to include extended-range formats with increased bit depth and enhanced color component sampling, scalable coding, and 3-D/stereo/multi-view video coding (the latter including the encoding of depth maps for use with advanced 3-D displays).

VII. CONCLUSION

The emerging HEVC standard has been developed and standardized collaboratively by both the ITU-T VCEG and ISO/IEC MPEG organizations. HEVC represents a number of advances in video coding technology. Its video coding layer design is based on conventional block-based motion-compensated hybrid video coding concepts, but with some important differences relative to prior standards.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000141

When used well together, the features of the new design provide approximately a 50% bit-rate savings for equivalent perceptual quality relative to the performance of prior standards (especially for a high-resolution video). For more details on compression performance, please refer to [23]. Implementation complexity analysis is outside the scope of this paper; however, the decoder implementation complexity of HEVC overall is not a major burden (e.g., relative to H.264/MPEG-4 AVC) using modern processing technology, and encoder complexity is also manageable. For more details on implementation complexity, please refer to [24].

Further information on and documents from the project are available in the JCT-VC document management system (http://phenix.int-evry.fr/jct/).

ACKNOWLEDGMENT

The authors would like to thank the experts of ITU-T VCEG, ISO/IEC MPEG, and the ITU-T/ISO/IEC Joint Collaborative Team on Video Coding for their contributions.

REFERENCES

[1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 9*, document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct. 2012.

[2] *Video Codec for Audiovisual Services at px64 kbit/s*, ITU-T Rec. H.261, version 1: Nov. 1990, version 2: Mar. 1993.

[3] *Video Coding for Low Bit Rate Communication*, ITU-T Rec. H.263, Nov. 1995 (and subsequent editions).

[4] *Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s—Part 2: Video*, ISO/IEC 11172-2 (MPEG-1), ISO/IEC JTC 1, 1993.

[5] *Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC 14496-2 (MPEG-4 Visual version 1), ISO/IEC JTC 1, Apr. 1999 (and subsequent editions).

[6] *Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video*, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG 2 Video), ITU-T and ISO/IEC JTC 1, Nov. 1994.

[7] *Advanced Video Coding for Generic Audio-Visual Services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC JTC 1, May 2003 (and subsequent editions).

[8] H. Samet, "The quadtree and related hierarchical data structures," *Comput. Survey*, vol. 16, no. 2, pp. 187–260, Jun. 1984.

[9] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[10] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 645–656, Jul. 2003.

[11] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 657–673, Jul. 2003.

[12] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[13] D. Marpe, H. Schwarz, and T. Wiegand, "Context-adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[14] G. J. Sullivan, *Meeting Report for 26th VCEG Meeting*, ITU-T SG16/Q6 document VCEG-Z01, Apr. 2005.

[15] *Call for Evidence on High-Performance Video Coding (HVC)*, MPEG document N10553, ISO/IEC JTC 1/SC 29/WG 11, Apr. 2009.

[16] *Joint Call for Proposals on Video Compression Technology*, ITU-T SG16/Q6 document VCEG-AM91 and ISO/IEC MPEG document N11113, ITU-T and ISO/IEC JTC 1, Jan. 2010.

[17] *Test Model Under Consideration*, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-A205, ITU-T and ISO/IEC JTC 1, Apr. 2010.

[18] T. Wiegand, J.-R. Ohm, G. J. Sullivan, W.-J. Han, R. Joshi, T. K. Tan, and K. Ugur, "Special section on the joint call for proposals on High Efficiency Video Coding (HEVC) standardization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1661–1666, Dec. 2010.

[19] K. McCann, *Tool Experiment 12: Evaluation of TMuC Tools*, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-B312, Jul. 2010.

[20] K. McCann, *Summary of Evaluation of TMuC Tools in TE12*, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-C225, Oct. 2010.

[21] K. McCann, B. Bross, S. Sekiguchi, and W.-J. Han, *Encoder-Side Description of HEVC Test Model (HM)*, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-C402, Oct. 2010.

[22] T. Wiegand, W.-J. Han, B. Bross, J.-R. Ohm, and G. J. Sullivan, *Working Draft 1 of High Efficiency Video Coding*, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-C402, Oct. 2010.

[23] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—Including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1668–1683, Dec. 2012.

[24] F. Bossen, B. Bross, K. Sühring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1684–1695, Dec. 2012.



Gary J. Sullivan (S'83–M'91–SM'01–F'06) received the B.S. and M.Eng. degrees in electrical engineering from the University of Louisville, Louisville, KY, in 1982 and 1983, respectively, and the Ph.D. and Engineers degrees in electrical engineering from the University of California, Los Angeles, in 1991.

He has been a long-standing Chairman or Co-Chairman of various video and image coding standardization activities in ITU-T and ISO/IEC. He is best known for leading the development of the ITU-T H.264|ISO/IEC 14496-10 MPEG-4 Advanced Video Coding standard and its Scalable Video Coding and 3-D/Stereo/Multiview Video Coding extensions. He is a Video/Image Technology Architect with the Windows Division, Microsoft Corporation. With Microsoft Corporation, he has been the Originator and Lead Designer of the DirectX Video Acceleration video decoding feature of the Microsoft Windows operating system. His current research interests and areas of publication include image and video compression, rate-distortion optimization, motion estimation and compensation, scalar and vector quantization, and loss resilient video coding.

Dr. Sullivan has received the IEEE Masaru Ibuka Consumer Electronics Award, the IEEE Consumer Electronics Engineering Excellence Award, the INCITS Technical Excellence Award, the IMTC Leadership Award, and the University of Louisville J. B. Speed Professional Award in Engineering. The team efforts that he has led have been recognized by the ATAS Primetime Emmy Engineering Award and by a pair of NATAS Technology and Engineering Emmy Awards. He is a fellow of the SPIE.



Jens-Rainer Ohm (M'92) received the Dipl.-Ing., Dr.-Ing., and Habil. degrees from the Technical University of Berlin (TUB), Berlin, Germany, in 1985, 1990, and 1997, respectively.

From 1985 to 1995, he was a Research Associate with the Institute of Telecommunication, TUB. From 1992 to 2000, he has served as a Lecturer on digital image processing, coding and transmission at TUB. From 1996 to 2000, he was a Project Coordinator with the Heinrich Hertz Institute, Berlin. In 2000, he became a Full Professor and since then has been the Chair of the Institute of Communication Engineering, RWTH Aachen University, Aachen, Germany. His current research and teaching activities cover the areas of motion-compensated, stereoscopic and 3-D image processing, multimedia signal coding, transmission and content description, audio signal analysis, as well as fundamental topics of signal processing and digital

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000142

communication systems. He has authored textbooks on multimedia signal processing, analysis and coding, on communication engineering and signal transmission, as well as numerous papers in the fields mentioned above.

Dr. Ohm has been participating in the work of the Moving Picture Experts Group (MPEG) since 1998. He has been the Chair and Co-Chair of various standardization activities in video coding, namely, the MPEG Video Subgroup since 2002, the Joint Video Team of MPEG and ITU-T SG 16 VCEG from 2005 to 2009, and currently, the Joint Collaborative Team on Video Coding as well as the Joint Collaborative Team on 3-D Video Coding Extensions. For T-CSVT, he has served as an Associate Editor from 2001 to 2005. He is a member of various professional organizations, including VDE/ITG, EURASIP, and AES.



Woo-Jin Han (M'02) received the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1997 and 2002, respectively.

He is currently a Professor with the Department of Software Design and Management, Gachon University, Seongnam, Korea. From 2003 to 2011, he was a Principal Engineer with the Multimedia Platform Laboratory, Digital Media and Communication Research and Development Center, Samsung Electronics, Suwon, Korea. Since 2003, he has contributed successfully to the ISO/IEC Moving Pictures Experts Group, Joint Video Team, and Joint Collaborative Team on Video Coding standardization efforts. His current research interests include high-efficiency video compression techniques, scalable video coding, multiview synthesis, and visual contents understanding.

Dr. Han was an editor of the HEVC video coding standard in 2010.



Thomas Wiegand (M'05–SM'08–F'11) received the Dipl.-Ing. degree in electrical engineering from the Technical University of Hamburg-Harburg, Germany, in 1995, and the Dr.-Ing. degree from the University of Erlangen-Nuremberg, Germany, in 2000.

He is currently a Professor with the Department of Electrical Engineering and Computer Science, Berlin Institute of Technology, Berlin, Germany, chairing the Image Communication Laboratory, and is jointly heading the Image Processing Department, Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin. He joined the Heinrich Hertz Institute in 2000 as the Head of the Image Communication Group, Image Processing Department. Since 1995, he has been an active participant in standardization for multimedia with successful submissions to ITU-T VCEG, ISO/IEC MPEG, 3GPP, DVB, and IETF. His current research interests include video processing and coding, multimedia transmission, and computer vision and graphics.

Dr. Wiegand was a recipient of the SPIE VCIP Best Student Paper Award in 1998, and the Fraunhofer Award and the ITG Award of the German Society for Information Technology in 2004. The projects that he co-chaired for development of the H.264/MPEG-4 AVC standard have been recognized by the 2008 ATAS Primetime Emmy Engineering Award and by a pair of NATAS Technology and Engineering Emmy Awards. In 2009, he received the Innovations Award of the Vodafone Foundation, the EURASIP Group Technical Achievement Award, and the Best Paper Award of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. In 2010, he received the Eduard Rhein Technology Award. In 2011, he received the Best Paper Award of EURASIP and the Karl Heinz Beckurts Award. In October 2000, he was appointed as the Associated Rapporteur of ITU-T VCEG. In December 2001, he was appointed as the Associated Rapporteur/Co-Chair of the JVT. In February 2002, he was appointed as the editor of the H.264/MPEG-4 AVC video coding standard and its extensions (FRExt and SVC). From 2005 to 2009, he was the Co-Chair of MPEG Video. He is a recipient of the 2012 IEEE Masaru Ibuka Technical Field Award.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000143

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2011 년 5 월 26 일 (26.05.2011)

PCT

(10) 국제공개번호
WO 2011/062392 A2

(51) 국제특허분류:
H04N 7/32 (2006.01) *H04N 7/24* (2011.01)

(21) 국제출원번호: PCT/KR2010/007839

(22) 국제출원일: 2010 년 11 월 8 일 (08.11.2010)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:
10-2009-0111302 2009 년 11 월 18 일 (18.11.2009) KR

(71) 출원인 (US 을(를) 제외한 모든 지정국에 대하여): **에스케이텔레콤 주식회사 (SK TELECOM CO., LTD.)** [KR/KR]; 서울 중구 을지로 2 가 11 번지, 100-999 Seoul (KR).

(72) 발명자; 겸

(75) 발명자/출원인 (US 에 한하여): **김수년 (KIM, Sunyeon)** [KR/KR]; 서울 송파구 잠실동 잠실엘스아파트 158 동 2603 호, 138-220 Seoul (KR). **임정연 (LIM, Jeongyeon)** [KR/KR]; 경기도 성남시 분당구 정자동 102 한솔마을아파트 415 동 1902 호, 463-010 Gyeonggi-do (KR). **이규민 (LEE, Gyumin)** [KR/KR]; 경기도 성남시 분당구 금곡동 현대아리온오피스텔 308 호, 463-853 Gyeonggi-do (KR). **최재훈 (CHOI, Jachoon)** [KR/KR]; 경기도 성남시 분당구 수내동 로얄팬리스 하우스빌 A 동 2012 호, 463-854 Gyeonggi-do (KR). **김용구 (KIM, Yonggoo)** [KR/KR]; 서울시 서초구 반포본동 주공아파트 97 동 507 호, 137-813 Seoul (KR). **최윤식 (CHOE, Yoonsik)** [KR/KR]; 경기도 고양시 일산서구 일산 3 동 후곡마을 태영아파트 1704 동 501 호, 411-313 Gyeonggi-do (KR). **최영호 (CHOI, Yungho)** [KR/KR]; 경기도 안양시 동안구 관양동 1595-1 동우베스티움 501 호, 431-060 Gyeonggi-do (KR).

(74) 대리인: **이철희 (LEE, Chulhee)**; 서울 강남구 역삼동 647-13 동궁빌딩 5 층, 135-911 Seoul (KR).

(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

공개:

— 국제조사보고서 없이 공개하며 보고서 접수 후 이를 별도 공개함 (규칙 48.2(g))

(54) Title: METHOD AND APPARATUS FOR ENCODING/DECODING A MOTION VECTOR BY SELECTING A SET OF PREDICTED CANDIDATE MOTION VECTORS, AND METHOD AND APPARATUS FOR IMAGE ENCODING/DECODING USING THE SAME

(54) 발명의 명칭 : 후보 예측 움직임 벡터 집합 선택을 이용한 움직임 벡터 부호화/복호화 방법 및 장치와 그를 이용한 영상 부호화/복호화 방법 및 장치

(57) Abstract: The present invention relates to a method and apparatus for encoding/decoding a motion vector by selecting a set of predicted candidate motion vectors, and a method and apparatus for image encoding/decoding using the same. The method for encoding a motion vector according to the present invention comprises: selecting one set of predicted candidate motion vectors among plural sets of predicted candidate motion vectors by using information about motions of neighboring blocks of a current block; selecting one of the predicted candidate motion vectors in the selected set of predicted candidate motion vectors as a predicted motion vector; encoding a differential vector which represents the difference between the current motion vector as a motion vector of the current block and the predicted motion vector being selected; and encoding a predicted motion vector index that represents the predicted motion vector being selected. According to the present invention, as a motion vector is encoded by selecting an efficient set of predicted candidate motion vectors, the magnitude of a differential vector to be encoded can be reduced without necessarily encoding additional information to indicate which set of predicted candidate motion vectors has been selected. In result, the compression efficiency of a motion vector and further the compression efficiency of an image can be improved.

(57) 요약서: 본 발명은 후보 예측 움직임 벡터 집합 선택을 이용한 움직임 벡터 부호화/복호화 방법 및 장치와 영상 부호화/복호화 방법 및 장치에 관한 것이다. 본 발명은 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하고, 선택되는 후보 예측 움직임 벡터 집합 내의 후보 예측 움직임 벡터 중에서 하나의 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하며, 현재 블록의 움직임 벡터인 현재 움직임 벡터와 선택되는 예측 움직임 벡터의 차이인 차분 벡터를 부호화하며, 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화하는 단계를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 방법을 제공한다. 본 발명에 의하면, 효율적인 후보 예측 움직임 벡터 집합을 선택하여 움직임 벡터를 부호화함에 따라 부호화할 차분 벡터의 크기를 줄이면서도 어떠한 후보 예측 움직임 벡터 집합이 선택되었는지를 나타내기 위한 부가 정보를 부호화하지 않아도 되므로, 움직임 벡터의 압축 효율을 향상시켜 결과적으로 영상의 압축 효율을 향상시킬 수 있다.

WO 2011/062392 A2

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000144

명세서

발명의 명칭: 후보 예측 움직임 벡터 집합 선택을 이용한 움직임 벡터 부호화/복호화 방법 및 장치와 그를 이용한 영상 부호화/복호화 방법 및 장치

기술분야

[1] 본 발명은 후보 예측 움직임 벡터 집합 선택을 이용한 움직임 벡터 부호화/복호화 방법 및 장치와 그를 이용한 영상 부호화/복호화 방법 및 장치에 관한 것이다. 더욱 상세하게는, 영상을 예측 부호화하는 데 이용되는 움직임 벡터를 효율적으로 압축하여 영상의 압축 효율을 향상시키기 위한 방법 및 장치에 관한 것이다.

배경기술

[2] H.264/AVC와 같은 통상적인 영상 압축 기술에서는 블록 기반으로 움직임을 추정하여 얻어지는 움직임 벡터를 예측 부호화하기 위해서, 부호화하고자 하는 블록의 주변 블록의 움직임 벡터들의 중앙값(Median)을 이용하여 예측 움직임 벡터를 결정하고, 부호화하고자 하는 움직임 벡터와 예측 움직임 벡터의 차분을 가변 길이 부호화하여 움직임 벡터를 압축한다.

[3] 최근 ITU-T VCEG(Video Coding Expert Group)에서는 KTA(Key Technical Area)라는 이름으로 기존의 H.264/AVC 기반으로 더 높은 성능을 갖는 코덱을 연구해왔는데, 그 중 MVComp(Competition-based Motion Vector Coding)라는 기법을 통하여 기존의 H.264/AVC의 움직임 벡터 부호화 기법을 개선하였다.

[4] MVComp는 여러 개의 후보 예측 움직임 벡터를 두고, 현재 움직임 벡터와 예측 후에 얻어지는 움직임 벡터와의 차분 값이 최소가 되는 후보 예측 움직임 벡터를 선택하여 복호화기에 선정된 후보 예측 움직임 벡터에 대한 정보를 전송해주는 방법으로, 기존의 H.264/AVC에 비해서 5% 정도의 부호화 압축 효율을 개선하였다. 하지만, 이 기법은 후보 예측 움직임 벡터의 개수가 늘어날수록 복호화기에 전송해야 할 인덱싱 부가 정보(Indexing Side Information)의 양이 증가하는 문제점이 있다.

[5] 이에, 여러 개의 후보 예측 움직임 벡터 중에 현재 움직임 벡터와 가장 유사할 것과 같은 하나의 움직임 벡터를 부호화기에서 선택하여, 선택된 움직임 벡터가 최적의 움직임 벡터인지 아닌지만을 구별하기 위한 부가 정보를 보내는 기술 등이 제안되었다. 하지만, 제안된 방법은 현재 프레임의 비트를 에러 없이 전달받았음에도 불구하고, 이전 프레임에서 발생한 에러 때문에 현재 프레임 및 다음 인트라 프레임 전까지의 프레임을 복원할 수 없게 되는 복호화기 충돌 문제(Decoder Crash Problem)와 복호화기에 많은 연산량 증가를 가져오는 한계를 가지는 문제점이 있다.

[6] 이에, 현재 부호화할 블록 주위의 움직임 벡터를 이용하여 현재 블록의 움직임

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000145

벡터를 결정하는 기법들이 제안되었다. 이 기법은 주변 블록의 움직임 벡터를 이용하여 인덱싱 부가 정보를 효율적으로 전송하고 후보 예측 움직임 벡터의 선택을 이전 프레임의 정보와 분리시켜 복호화기 충돌 문제를 해결하였지만, 한정된 개수의 후보 예측 움직임 벡터를 사용하기 때문에 압축 성능에 있어서 제한적이라는 한계를 가지는 문제점이 있다.

발명의 상세한 설명

기술적 과제

[7] 전술한 문제점을 해결하기 위해 본 발명은, 부호화할 움직임 벡터와 더욱 유사한 예측 움직임 벡터를 이용하여 움직임 벡터를 부호화하면서도 그로 인해 발생하는 비트량을 줄여 움직임 벡터를 효율적으로 압축하고 그에 따라 영상의 압축 효율을 향상시키는 데 주된 목적이 있다.

과제 해결 수단

[8] 전술한 목적을 달성하기 위해 본 발명은, 움직임 벡터를 부호화하는 장치에 있어서, 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하는 후보 예측 움직임 벡터 집합 선택기; 선택되는 후보 예측 움직임 벡터 집합 내의 후보 예측 움직임 벡터 중에서 하나의 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하는 예측 움직임 벡터 선택기; 현재 블록의 움직임 벡터인 현재 움직임 벡터와 선택되는 예측 움직임 벡터의 차이인 차분 벡터를 부호화하는 차분 벡터 부호화기; 및 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화하는 예측 움직임 벡터 부호화기를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 장치를 제공한다.

[9] 또한, 본 발명의 다른 목적에 의하면, 영상을 부호화하는 장치에 있어서, 현재 블록의 현재 움직임 벡터를 결정하고, 결정되는 현재 움직임 벡터를 이용하여 현재 블록을 예측 부호화하는 영상 부호화기; 및 복수 개의 후보 예측 움직임 벡터 집합 중 현재 블록의 주변 블록의 움직임 정보를 기초로 선택되는 후보 예측 움직임 벡터 집합에서 예측 움직임 벡터를 선택하고, 현재 움직임 벡터와 선택되는 예측 움직임 벡터의 차이인 차분 벡터 및 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화하는 움직임 벡터 부호화기를 포함하는 것을 특징으로 하는 영상 부호화 장치를 제공한다.

[10] 또한, 본 발명의 또 다른 목적에 의하면, 움직임 벡터를 복호화하는 장치에 있어서, 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하는 후보 예측 움직임 벡터 집합 선택기; 움직임 벡터 데이터로부터 추출되는 인덱스 데이터를 복호화하여 예측 움직임 벡터 인덱스를 복원하고, 선택되는 후보 예측 움직임 벡터 집합에서 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 복원하는 예측 움직임 벡터 복원기;

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000146

및 움직임 벡터 데이터로부터 추출되는 차분 벡터 데이터를 복호화하여 차분 벡터를 복원하고, 복원되는 차분 벡터와 복원되는 예측 움직임 벡터를 가산하여 현재 블록의 현재 움직임 벡터를 복원하는 현재 움직임 벡터 복원기를 포함하는 것을 특징으로 하는 움직임 벡터 복호화 장치를 제공한다.

[11] 또한, 본 발명의 또 다른 목적에 의하면, 영상을 복호화하는 장치에 있어서, 비트스트림으로부터 추출되는 움직임 벡터 데이터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원하고, 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하며, 선택되는 후보 예측 움직임 벡터 집합에서 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하며, 복원되는 차분 벡터와 선택되는 예측 움직임 벡터를 가산하여 현재 블록의 현재 움직임 벡터를 복원하는 움직임 벡터 복호화기; 및 비트스트림으로부터 추출되는 영상 데이터를 복원되는 현재 움직임 벡터를 이용하여 예측 복호화하여 현재 블록을 복원하는 영상 복호화기를 포함하는 것을 특징으로 하는 영상 복호화 장치를 제공한다.

[12] 또한, 본 발명의 또 다른 목적에 의하면, 움직임 벡터를 부호화하는 방법에 있어서, 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계; 선택되는 후보 예측 움직임 벡터 집합 내의 후보 예측 움직임 벡터 중에서 하나의 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하는 단계; 현재 블록의 움직임 벡터인 현재 움직임 벡터와 선택되는 예측 움직임 벡터의 차이인 차분 벡터를 부호화하는 단계; 및 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화하는 단계를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 방법을 제공한다.

[13] 또한, 본 발명의 또 다른 목적에 의하면, 영상을 부호화하는 방법에 있어서, 현재 블록의 현재 움직임 벡터를 결정하는 단계; 결정되는 현재 움직임 벡터를 이용하여 현재 블록을 예측 부호화하는 단계; 복수 개의 후보 예측 움직임 벡터 집합 중 현재 블록의 주변 블록의 움직임 정보를 기초로 선택되는 후보 예측 움직임 벡터 집합에서 예측 움직임 벡터를 선택하는 단계; 및 현재 움직임 벡터와 선택되는 예측 움직임 벡터의 차이인 차분 벡터와 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화하는 단계를 포함하는 것을 특징으로 하는 영상 부호화 방법을 제공한다.

[14] 또한, 본 발명의 또 다른 목적에 의하면, 움직임 벡터를 복호화하는 방법에 있어서, 움직임 벡터 데이터로부터 추출되는 차분 벡터 데이터와 인덱스 데이터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원하는 단계; 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계; 선택되는 후보 예측 움직임 벡터 집합에서 복원되는 예측 움직임 벡터 인덱스에

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000147

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 00148

의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 복원하는 단계; 및 복원되는 차분 벡터와 복원되는 예측 움직임 벡터를 가산하여 현재 블록의 현재 움직임 벡터를 복원하는 단계를 포함하는 것을 특징으로 하는 움직임 벡터 복호화 방법을 제공한다.

[15] 또한, 본 발명의 또 다른 목적에 의하면, 영상을 복호화하는 방법에 있어서, 비트스트림으로부터 추출되는 움직임 벡터 데이터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원하는 단계; 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계; 선택되는 후보 예측 움직임 벡터 집합에서 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하는 단계; 복원되는 차분 벡터와 선택되는 예측 움직임 벡터를 가산하여 현재 블록의 현재 움직임 벡터를 복원하는 단계; 및 비트스트림으로부터 추출되는 영상 데이터를 복원되는 현재 움직임 벡터를 이용하여 예측 복호화하여 현재 블록을 복원하는 단계를 포함하는 것을 특징으로 하는 영상 복호화 방법을 제공한다.

발명의 효과

[16] 이상에서 설명한 바와 같이 본 발명에 의하면, 효율적인 후보 예측 움직임 벡터 집합을 선택하여 움직임 벡터를 부호화함에 따라 부호화할 차분 벡터의 크기를 줄이면서도 어떠한 후보 예측 움직임 벡터 집합이 선택되었는지를 나타내기 위한 부가 정보를 부호화하지 않아도 되므로, 움직임 벡터의 압축 효율을 향상시켜 결과적으로 영상의 압축 효율을 향상시킬 수 있다.

도면의 간단한 설명

[17] 도 1은 본 발명의 일 실시예에 따른 영상 부호화 장치를 간략하게 나타낸 블록 구성도,

[18] 도 2는 본 발명의 일 실시예에 따른 움직임 벡터 부호화 장치를 간략하게 나타낸 블록 구성도,

[19] 도 3은 본 발명의 일 실시예에 따른 주변 블록의 움직임 정보를 나타낸 예시도,

[20] 도 4는 본 발명의 일 실시예에 따라 후보 예측 움직임 벡터 집합을 선택하는 방법을 구현한 예를 나타낸 예시도,

[21] 도 5는 본 발명의 일 실시예에 따른 움직임 벡터 부호화 방법을 설명하기 위한 순서도,

[22] 도 6은 본 발명의 일 실시예에 따른 영상 부호화 방법을 설명하기 위한 순서도,

[23] 도 7은 본 발명의 일 실시예에 따른 영상 복호화 장치를 간략하게 나타낸 블록 구성도,

[24] 도 8은 본 발명의 일 실시예에 따른 움직임 벡터 복호화 장치를 간략하게 나타낸 블록 구성도,

[25] 도 9는 본 발명의 일 실시예에 따른 움직임 벡터 복호화 방법을 설명하기 위한

순서도,

[26] 도 10은 본 발명의 일 실시예에 따른 영상 복호화 방법을 설명하기 위한 순서도이다.

발명의 실시를 위한 형태

[27] 이하, 본 발명의 일부 실시예들을 예시적인 도면을 통해 상세하게 설명한다. 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 또한, 본 발명을 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 발명의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다.

[28] 또한, 본 발명의 구성 요소를 설명하는 데 있어서, 제 1, 제 2, A, B, (a), (b) 등의 용어를 사용할 수 있다. 이러한 용어는 그 구성 요소를 다른 구성 요소와 구별하기 위한 것일 뿐, 그 용어에 의해 해당 구성 요소의 본질이나 차례 또는 순서 등이 한정되지 않는다. 어떤 구성 요소가 다른 구성요소에 "연결", "결합" 또는 "접속"된다고 기재된 경우, 그 구성 요소는 그 다른 구성요소에 직접적으로 연결되거나 접속될 수 있지만, 각 구성 요소 사이에 또 다른 구성 요소가 "연결", "결합" 또는 "접속"될 수도 있다고 이해되어야 할 것이다.

[29] 이하에서 후술할 움직임 벡터 부호화 장치(Motion Vector Encoding Apparatus), 움직임 벡터 복호화 장치(Motion Vector Decoding Apparatus), 영상 부호화 장치(Video Encoding Apparatus), 영상 복호화 장치(Video Decoding Apparatus)는 개인용 컴퓨터(PC: Personal Computer), 노트북 컴퓨터, 개인 휴대 단말기(PDA: Personal Digital Assistant), 휴대형 멀티미디어 플레이어(PMP: Portable Multimedia Player), 플레이스테이션 포터블(PSP: PlayStation Portable), 이동통신 단말기(Mobile Communication Terminal) 등과 같은 사용자 단말기이거나 응용 서버와 서비스 서버 등 서버 단말기일 수 있으며, 각종 기기 또는 유무선 통신망과 통신을 수행하기 위한 통신 모뎀 등의 통신 장치, 움직임 벡터를 부호화거나 복호화거나 영상을 부호화하거나 복호화하기 위한 각종 프로그램과 데이터를 저장하기 위한 메모리, 프로그램을 실행하여 연산 및 제어하기 위한 마이크로프로세서 등을 구비하는 다양한 장치를 의미한다.

[30] 또한, 움직임 벡터 부호화 장치 또는 영상 부호화 장치에 의해 비트스트림으로 부호화된 움직임 벡터 또는 영상은 실시간 또는 비실시간으로 인터넷, 근거리 무선 통신망, 무선랜망, 와이브로망, 이동통신망 등의 유무선 통신망 등을 통하거나 케이블, 범용 직렬 버스(USB: Universal Serial Bus) 등과 같은 다양한 통신 인터페이스를 통해 움직임 벡터 복호화 장치 또는 영상 복호화 장치로 전송되어 움직임 벡터 복호화 장치에서 복호화되어 움직임 벡터로서 복원되거나 영상 복호화 장치에서 복호화되어 영상으로 복원되고 재생될 수 있다.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000149

[31] 통상적으로 동영상은 일련의 픽처(Picture)로 구성되어 있으며, 각 픽처들은 블록(Block)과 같은 소정의 영역으로 분할된다. 영상의 영역이 블록으로 분할되는 경우에는 분할된 블록은 부호화 방법에 따라 크게 인트라 블록(Intra Block), 인터 블록(Inter Block)으로 분류된다. 인트라 블록은 인트라 예측 부호화(Intra Prediction Coding) 방식을 사용하여 부호화되는 블록을 뜻하는데, 인트라 예측 부호화란 현재 부호화를 수행하는 현재 픽처 내에서 이전에 부호화되고 복호화되어 복원된 블록들의 화소를 이용하여 현재 부호화하고자 하는 블록인 현재 블록의 화소를 예측함으로써 예측 블록을 생성하고 현재 블록의 화소와의 차분값을 부호화하는 방식이다. 인터 블록은 인터 예측 부호화(Inter Prediction Coding)를 사용하여 부호화되는 블록을 뜻하는데, 인터 예측 부호화란 하나 이상의 과거 픽처 또는 미래 픽처를 참조하여 현재 픽처 내의 현재 블록을 예측함으로써 예측 블록을 생성하고 현재 블록과의 차분값을 부호화하는 방식이다. 여기서, 현재 픽처를 부호화하거나 복호화하는데 참조되는 픽처를 참조 픽처(Reference Picture)라고 한다.

[32] 도 1은 본 발명의 일 실시예에 따른 영상 부호화 장치를 간략하게 나타낸 블록 구성도이다.

[33] 본 발명의 일 실시예에 따른 영상 부호화 장치(100)는 영상을 부호화하는 장치로서, 움직임 벡터 부호화기(Motion Vector Encoder, 110)와 영상 부호화기(Video Encoder, 120)를 포함하여 구성될 수 있다.

[34] 움직임 벡터 부호화기(110)는 복수 개의 후보 예측 움직임 벡터 집합(Candidate Predicted Motion Vector Set) 중 현재 블록의 주변 블록(Neighbor Block)의 움직임 정보(Motion Information)를 기초로 선택되는 후보 예측 움직임 벡터 집합에서 예측 움직임 벡터(Predicted Motion Vector)를 선택하고, 현재 움직임 벡터(Current Motion Vector)와 선택되는 예측 움직임 벡터의 차이인 차분 벡터(Differential Motion Vector) 및 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스(Predicted Motion Vector Index)를 부호화한다. 차분 벡터는 부호화되어 차분 벡터 데이터로서 생성되고 예측 움직임 벡터 인덱스는 부호화되어 인덱스 데이터로서 생성된다. 따라서, 움직임 벡터 부호화기(110)는 차분 벡터 데이터와 그룹 인덱스 데이터를 포함하는 움직임 벡터 데이터를 생성한다.

[35] 여기서, 움직임 벡터 부호화기(110)는 차분 벡터를 생성하는 데 있어서, 영상 부호화기(120)에서 현재 블록을 예측 부호화하기 위해 결정된 현재 움직임 벡터를 이용한다. 움직임 벡터 부호화기(110)에 대해서는 후술하는 과정에서 도 2를 통해 상세히 설명한다.

[36] 영상 부호화기(120)는 현재 블록의 움직임 벡터인 현재 움직임 벡터를 결정하고 현재 움직임 벡터를 이용하여 현재 블록을 예측 부호화한다. 이와 같이 현재 블록이 예측 부호화되어 영상 데이터가 생성된다.

[37] 이를 위해, 영상 부호화기(120)는 예측기(Predictor), 감산기(Subtracter), 변환기 및 양자화기(Transformer and Quantizer), 부호화기(Encoder)를 포함하여 구성될

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000150

수 있으며, 역 양자화기 및 역 변환기(Inverse Transformer and Inverse Quantizer), 가산기(Adder), 디블로킹 필터(Deblocking Filter), 메모리(Memory) 등을 추가로 포함할 수 있다. 여기서, 예측기는 현재 블록의 움직임을 추정하여 현재 움직임 벡터를 결정하고 현재 움직임 벡터를 이용하여 현재 블록의 움직임을 보상하여 예측 블록(Predicted Block)을 생성하며, 감산기는 현재 블록과 예측 블록을 감산하여 잔여 블록(Residual Block)을 생성하며, 변환기 및 양자화기는 잔여 블록을 변환 및 양자화하여 양자화된 변환 계수(Quantized Transform Coefficient)를 생성하며, 부호화기는 양자화된 변환 계수를 부호화하여 영상 데이터를 생성한다. 또한, 역 양자화기 및 역 변환기는 양자화된 변환 계수를 역 양자화 및 역 변환하여 잔여 블록을 복원하고 가산기는 예측 블록과 복원되는 잔여 블록을 복원하여 현재 블록을 복원하며, 복원된 현재 블록은 디블로킹 필터(Deblocking Filter)에 의해 디블로킹 필터링되어 메모리에 픽처 단위로 누적되어 참조 픽처로서 저장되고 다음 블록 또는 다음 픽처를 예측하는 데 활용된다.

[38] 도 2는 본 발명의 일 실시예에 따른 움직임 벡터 부호화 장치를 간략하게 나타낸 블록 구성도이다.

[39] 본 발명의 일 실시예에 따른 움직임 벡터 부호화 장치는 도 1을 통해 전술한 영상 부호화 장치(100)에서는 움직임 벡터 부호화기(110)로 구현될 수 있다. 이하에서는 설명의 편의를 위해, 본 발명의 일 실시예에 따른 움직임 벡터 부호화 장치를 움직임 벡터 부호화기(110)라 칭한다.

[40] 움직임 벡터 부호화기(110)는 후보 예측 움직임 벡터 집합 선택기(Candidate Predicted Motion Vector Set Selector, 210), 예측 움직임 벡터 선택기(Predicted Motion Vector Selector, 220), 차분 벡터 부호화기(Differential Motion Vector Encoder, 230) 및 예측 움직임 벡터 부호화기(Predicted Motion Vector Encoder, 240)를 포함하여 구성될 수 있다.

[41] 후보 예측 움직임 벡터 집합 선택기(210)는 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택한다.

[42] 본 발명의 일 실시예에서, 현재 블록의 주변 블록이란 현재 블록이 포함된 현재 픽처에서 현재 블록이 부호화되기 전에 이미 부호화되고 복호화되어 기 복원된 블록 중 현재 블록의 주변에 위치한 블록을 말한다. 주변 블록은 현재 블록과 인접한 인접 블록이 될 수도 있지만 반드시 인접한 블록만으로 한정되지는 않는다. 주변 블록의 움직임 정보란 기 부호화되고 복호화되어 복원된 주변 블록에 대해 움직임 벡터와 관련하여 획득된 정보들을 말하며, 주변 블록의 움직임 벡터와 예측 움직임 벡터 등이 될 수 있다.

[43] 도 3은 본 발명의 일 실시예에 따른 주변 블록의 움직임 정보를 나타낸 예시도이다.

[44] 3A는 현재 블록의 주변 블록과 주변 블록의 움직임 벡터를 예시적으로

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000151

나타내었다. 현재 블록의 주변에 위치한 블록들 중에서 현재 블록의 왼쪽에 인접한 블록을 블록 A, 위쪽에 인접한 블록을 블록 B, 위쪽 오른쪽에 인접한 블록을 블록 C라고 가정하면, 블록 A, 블록 B 및 블록 C가 현재 블록의 주변 블록이 될 수 있다. 이때, 블록 A, 블록 B 및 블록 C는 모두 현재 블록이 부호화되기 전에 이미 부호화되고 복호화되어 복원된 블록들이다. 3A에서는 블록 A, 블록 B 및 블록 C만이 현재 블록의 주변 블록인 것으로 도시하였지만, 이에 한정되지 않고 현재 블록의 위쪽 왼쪽에 위치한 블록 등과 같이 다른 블록들도 주변 블록으로 이용될 수 있다.

[45] 3A에서, MV_a, MV_b, MV_c는 블록 A, 블록 B 및 블록 C 각각의 움직임 벡터를 나타낸다. 블록 A, 블록 B 및 블록 C의 움직임 벡터는 각 블록을 부호화할 때 이미 결정되어 영상 부호화 장치(100) 또는 움직임 벡터 부호화기(110)의 버퍼 또는 메모리 등에 저장되어 있으므로, 영상 부호화 장치(100), 움직임 벡터 부호화기(110) 또는 후보 예측 움직임 벡터 집합 선택기(210)가 현재 블록을 부호화하거나 현재 블록의 움직임 벡터를 부호화할 때 언제든지 획득하여 이용할 수 있는 정보이다.

[46] 3B는 주변 블록의 예측 움직임 벡터를 예시적으로 나타내었다. 3B에서, PMV_a, PMV_b, PMV_c는 블록 A, 블록 B 및 블록 C 각각의 예측 움직임 벡터를 나타낸다. 블록 A, 블록 B 및 블록 C의 예측 움직임 벡터도 각 블록을 부호화할 때 이미 결정되어 영상 부호화 장치(100) 또는 움직임 벡터 부호화기(110)의 버퍼 또는 메모리 등에 저장되어 있으므로, 영상 부호화 장치(100), 움직임 벡터 부호화기(110) 또는 후보 예측 움직임 벡터 집합 선택기(210)가 현재 블록을 부호화하거나 현재 블록의 움직임 벡터를 부호화할 때 언제든지 획득하여 이용할 수 있는 정보이다.

[47] 따라서, 3A와 3B에 나타낸 예의 경우, 주변 블록의 움직임 정보는 3C에 나타낸 바와 같이 MV_a, MV_b, MV_c, PMV_a, PMV_b, PMV_c가 될 수 있다.

[48] 후보 예측 움직임 벡터 집합 선택기(210)는, 도 3에 도시한 바와 같이, 주변 블록의 움직임 벡터와 예측 움직임 벡터를 주변 블록의 움직임 정보로서 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택할 수 있다.

[49] 일 예로, 후보 예측 움직임 벡터 집합 선택기(210)는 주변 블록의 움직임 정보를 이용하여 카메라의 움직임 여부를 판단하고 판단되는 카메라의 움직임 여부를 기초로 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택할 수 있다. 이를 위해, 후보 예측 움직임 벡터 집합 선택기(210)는 주변 블록들 중에서 움직임 벡터가 제로 벡터(Zero Vector)인 주변 블록의 개수가 기 설정된 개수 이상인지 여부를 판단하여 카메라의 움직임 여부를 판단할 수 있다. 즉, 후보 예측 움직임 벡터 집합 선택기(210)는 움직임 벡터가 제로 벡터인 주변 블록의 개수가 기 설정된 개수 이상인 경우에는 카메라의 움직임이 없는 것으로 판단하고, 움직임 벡터가 제로 벡터인 주변

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000152

블록의 개수가 기 설정된 개수 미만인 경우에는 카메라의 움직임이 있는 것으로 판단할 수 있다.

[50] 도 3에 도시한 예에서, 주변 블록들의 움직임 벡터 MV_a, MV_b, MV_c가 각각 (0, 0), (0, 0), (0, 1)이고, 카메라의 움직임 여부를 판단하기 위한 개수로 2 개가 설정되었으며, 복수 개의 후보 예측 움직임 벡터 집합으로서 $\{MV_a, MV_{H.264}\}$, $\{MV_a, MV_{extspa}\}$ 두 개의 후보 예측 움직임 벡터 집합이 설정되었다고 가정하면, 움직임 벡터가 제로 벡터인 주변 블록의 개수가 2로서 기 설정된 개수인 2 이상이므로, 카메라의 움직임이 없는 것으로 판단되어 $MV_{H.264}$가 포함된 $\{MV_a, MV_{H.264}\}$가 후보 예측 움직임 벡터 집합으로서 선택될 수 있다. 이때, 주변 블록들의 움직인 벡터 MV_a, MV_b, MV_c가 각각 (0, 0), (1, 0), (0, 1)이라고 가정하면, 움직임 벡터가 제로 벡터인 주변 블록의 개수가 1로서 기 설정된 개수인 2 미만이므로, 카메라의 움직임이 있는 것으로 판단되어 MV_{extspa}가 포함된 $\{MV_a, MV_{extspa}\}$가 후보 예측 움직임 벡터 집합으로서 선택될 수 있다.

[51] 이때, 카메라의 움직임이 없는 것으로 판단되면 $MV_{H.264}$가 포함된 집합을 후보 예측 움직임 벡터 집합으로서 선택하고 카메라의 움직임이 있는 것으로 판단되면 MV_{extspa}가 포함된 집합을 후보 예측 움직임 벡터 집합으로서 선택하는 것은 예시적인 것일 뿐, 반드시 $MV_{H.264}$ 또는 MV_{extspa}가 포함된 집합을 선택해야 하는 것은 아니며, 어떠한 후보 예측 움직임 벡터가 포함된 집합을 선택해야 하는 지는 실험을 통해 경험적으로 결정될 수 있다. 즉, 카메라의 움직임 여부에 따라 선택되어 후보 예측 움직임 벡터 집합에 포함될 후보 예측 움직임 벡터는 실험을 통해 경험적으로 결정될 수 있으며, 이와 같이 경험적으로 결정되는 데이터는 영상 부호화 장치(110)와 영상 복호화 장치에 미리 설정하여 저장될 수 있다.

[52] 다른 예로, 후보 예측 움직임 벡터 집합 선택기(210)는 주변 블록의 움직임 정보를 이용하여 주변 블록의 객체 움직임(Object Motion)을 판단하고, 판단되는 주변 블록의 객체 움직임을 기초로 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택할 수 있다. 이를 위해, 후보 예측 움직임 벡터 집합 선택기(210)는 주변 블록의 예측 움직임 벡터를 이용하여 주변 블록의 객체 움직임을 판단할 수 있다. 즉, 후보 예측 움직임 벡터 집합 선택기(210)는 주변 블록의 객체 움직임을 판단하기 위해 주변 블록의 움직임 정보로서 주변 블록의 예측 움직임 벡터를 분석하고, 분석된 주변 블록의 예측 움직임 벡터의 분포에 따라 이용 가능한 후보 예측 움직임 벡터들 중 하나 이상의 후보 예측 움직임 벡터를 선택하고, 복수 개의 후보 예측 움직임 벡터 집합 중에서 선택된 하나 이상의 후보 예측 움직임 벡터를 포함하는 후보 예측 움직임 벡터 집합을 하나의 후보 예측 움직임 벡터 집합으로서 선택할 수 있다.

[53] 도 3에 도시한 예에서, 설정된 복수 개의 후보 예측 움직임 벡터 집합으로서 $\{MV_a, MV_{H.264}\}$, $\{MV_b, MV_{H.264}\}$, $\{MV_c, MV_{H.264}\}$ 세 개의 후보 예측 움직임 벡터 집합이 설정되었다고 가정하면, 주변 블록들의 예측 움직임 벡터 PMV_a, PMV_b,

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000153

PMV_c는 각각 MV_a, MV_a, MV_c인데, 이 경우 현재 블록의 예측 움직임 벡터는 MV_a가 될 확률이 MV_a나 MV_c가 될 확률보다 높다는 가정 하에서, MV_a, MV_a, MV_c, $MV_{H.264}$, MV_{extspa}등과 같은 이용 가능한 후보 예측 움직임 벡터들 중에서 MV_b를 선택하여, 선택된 MV_b를 포함하는 $\{MV_b, MV_{H.264}\}$가 후보 예측 움직임 벡터 집합으로서 선택될 수 있다. 이때, 주변 블록들의 예측 움직임 벡터 PMV_a, PMV_b, PMV_c가 각각 MV_a, MV_a, MV_c일 때, 현재 블록의 예측 움직임 벡터는 MV_b가 될 확률이 MV_a나 MV_c가 될 확률보다 높다는 것은 예시적인 것일 뿐, 이는 실험을 통해 경험적으로 결정될 수 있다. 즉, 주변 블록의 예측 움직임 벡터의 분포에 따라 선택되어 후보 예측 움직임 벡터 집합에 포함될 후보 예측 움직임 벡터는 실험을 통해 경험적으로 결정될 수 있으며, 이와 같이 경험적으로 결정되는 데이터는 영상 부호화 장치(110)와 영상 복호화 장치에 미리 설정하여 저장될 수 있다.

[54] 또 다른 예로, 도 4에 도시한 바와 같이, 선술한 카메라의 움직임 여부와 주변 블록의 객체 움직임 모두를 기초로 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택할 수 있다.

[55] 도 4는 본 발명의 일 실시예에 따라 후보 예측 움직임 벡터 집합을 선택하는 방법을 구현한 예를 나타낸 예시도이다.

[56] 복수 개의 후보 예측 움직임 벡터 집합들이 $\{MV_a, MV_{H.264}\}$, $\{MV_b, MV_{H.264}\}$, $\{MV_c, MV_{H.264}\}$, $\{MV_a, MV_{extspa}\}$, $\{MV_b, MV_{extspa}\}$, $\{MV_c, MV_{extspa}\}$로 설정된 경우(S410), 후보 예측 움직임 벡터 집합 선택기(210)는 주변 블록의 움직임 정보를 분석하여(S420), 주변 블록의 객체 움직임을 기초로 전술한 예에서와 같이 이용 가능한 후보 예측 움직임 벡터 중에서 MV_a를 선택하며(S430), 카메라 움직임이 없는지 여부를 판단하여(S410), 카메라 움직임이 없는 경우에는 $MV_{H.264}$를 선택하여(S450), $\{MV_a, MV_{H.264}\}$를 후보 예측 움직임 벡터 집합으로서 선택하며(S460), 카메라 움직임이 있는 경우에는 MV_{extspa}를 선택하여(S470), $\{MV_a, MV_{extspa}\}$를 후보 예측 움직임 벡터 집합으로서 선택할 수 있다(S480).

[57] 도 3 및 도 4를 통해서는 복수 개의 후보 예측 움직임 벡터 집합이 2 개, 3 개, 6 개인 경우에 대해서 예를 들었지만, 후보 예측 움직임 벡터 집합은 2 개 이상이라면 제한 없이 설정될 수 있다. 또한, 도 3 및 도 4를 통해서는 각 후보 예측 움직임 벡터 집합 내에 두 개의 후보 예측 움직임 벡터가 포함되는 것으로 예를 들었지만, 반드시 두 개에 한정되지 않고 두 개 이상 복수 개의 후보 예측 움직임 벡터가 포함될 수 있다.

[58] 다시 도 2를 참조하면, 예측 움직임 벡터 선택기(220)는 선택되는 후보 예측 움직임 벡터 집합 내의 후보 예측 움직임 벡터 중에서 하나의 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택한다. 이를 위해, 예측 움직임 벡터 선택기(220)는 후보 예측 움직임 벡터 집합 선택기(210)에 의해 선택되는 후보 예측 움직임 벡터 집합 내에 포함되는 후보 예측 움직임 벡터들 중에서 율-왜곡 측면에서 최적이 되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택할 수

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000154

있다. 여기서, 율-왜곡 측면에서 최적이 되는 후보 예측 움직임 벡터란 해당 후보 예측 움직임 벡터들 각각을 이용하여 현재 움직임 벡터를 부호화하고 현재 블록을 부호화하여 생성되는 비트스트림의 율-왜곡 비용이 가장 작은 후보 예측 움직임 벡터를 말한다.

[59] 예를 들어, 예측 움직임 벡터 선택기(220)에 의해 선택되는 후보 예측 움직임 벡터 집합이 $\{MV_a, MV_{H.264}\}$라고 가정하면, 수학식 1을 이용하여 예측 움직임 벡터가 선택될 수 있다.

[60] [수학식 1]

[61] $$BMVI = \arg\min_{l}[\sum_{i}\sum_{j}| x_{i,j} - \hat{x}_{i,j} | + \lambda \cdot \{R(MV_c - MV_l) + R(l)\}]$$

[62] 수학식 1에서, BMVI(Best Motion Vector Index)는 예측 움직임 벡터 선택기(220)에 의해 선택되는 예측 움직임 벡터를 식별하기 위한 예측 움직임 벡터 인덱스를 나타내며, $x_{i,y}$는 현재 화소의 좌표 (i, j)에 대한 화소값을 나타내며, $\hat{x}_{i,j}$는 참조 화소의 좌표 (i, j)에 대한 화소값을 나타내며, $R(MV\text{-}MVl)$과 $R(l)$은 각각 현재 블록의 움직임 벡터와 예측 움직임 벡터와의 차이를 부호화하는 데 소요되는 비트량과 예측 움직임 벡터 인덱스를 부호화하는 데 소요되는 비트량이 계산된 값을 나타낸다.

[63] 차분 벡터 부호화기(230)는 현재 블록의 움직임 벡터인 현재 움직임 벡터와 선택되는 예측 움직임 벡터의 차이인 차분 벡터를 부호화한다. 즉, 차분 벡터 부호화기(230)는 현재 움직임 벡터와 예측 움직임 벡터 선택기(220)에 의해 선택되는 예측 움직임 벡터를 감산하여 차분 벡터를 생성하고 차분 벡터를 부호화하여 차분 벡터 데이터를 생성한다. 예를 들어, 예측 움직임 벡터 선택기(220)에 의해 선택되는 예측 움직임 벡터가 MV_a라고 가정하면, 차분 벡터는 수학식 2와 같이 계산될 수 있다. 수학식 2에서, DMV는 차분 벡터를 나타내고, MV는 현재 움직임 벡터를 나타낸다.

[64] [수학식 2]

[65] $$DMV = MV_c - MV_a$$

[66] 다만, 차분 벡터 부호화기(230)는 차분 벡터를 별도로 부호화하지 않고 예측 움직임 벡터 선택기(220)가 율-왜곡 비용을 구하기 위해 차분 벡터를 구하고 부호화한 경우 예측 움직임 벡터 선택기(220)에 의해 부호화된 차분 벡터 데이터를 출력할 수 있다. 차분 벡터를 부호화하는 기법으로서는 고정 길이 부호화(Fixed Length Coding), 가변 길이 부호화(Variable Length Coding), 산술 부호화(Arithmetic Coding) 등과 같은 엔트로피 부호화(Entropy Coding) 기법이 이용될 수 있다.

[67] 예측 움직임 벡터 부호화기(240)는 예측 움직임 벡터 선택기(220)에 의해 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화한다. 예측 움직임 벡터 인덱스를 부호화하는 기법으로서는 차분 벡터를 부호화할

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000155

때와 마찬가지로 고정 길이 부호화, 가변 길이 부호화, 산술 부호화 등과 같은 엔트로피 부호화 기법이 이용될 수 있다.

[68] 예를 들어, 후보 예측 움직임 벡터 집합 선택기(210)에 의해 선택되는 후보 예측 움직임 벡터 집합이 $\{MV_a, MV_{H.264}\}$라고 가정하면, MV_a의 예측 움직임 벡터 인덱스는 '0'이고 $MV_{H.264}$의 예측 움직임 벡터 인덱스는 '1'로 설정될 수 있다. 따라서, 이 경우, 후보 예측 움직임 벡터 집합 내에 두 개의 후보 예측 움직임 벡터만이 존재하므로 예측 움직임 벡터 인덱스가 부호화되어 생성되는 인덱스 데이터는 '0' 또는 '1'의 값을 가지는 1 비트로 생성될 수 있다. 여기서, 영상 복호화 장치 또는 움직임 벡터 복호화 장치에서 예측 움직임 벡터 인덱스를 올바르게 복원하기 위해서는 후보 예측 움직임 벡터 집합별로 각 집합 내에 포함되는 후보 예측 움직임 벡터에 대한 예측 움직임 벡터 인덱스가 영상 부호화 장치(100)와 영상 복호화 장치 또는 움직임 벡터 부호화 장치(110)와 움직임 벡터 복호화 장치에 미리 동일하게 설정되어 저장되어야 한다.

[69] 도 5는 본 발명의 일 실시예에 따른 움직임 벡터 부호화 방법을 설명하기 위한 순서도이다.

[70] 본 발명의 일 실시예에 따른 움직임 벡터 부호화 방법에 따르면, 움직임 벡터 부호화기(110)는 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하고(S510), 선택되는 후보 예측 움직임 벡터 집합 내의 후보 예측 움직임 벡터 중에서 하나의 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하며(S520), 현재 블록의 움직임 벡터인 현재 움직임 벡터와 선택되는 예측 움직임 벡터의 차이인 차분 벡터를 부호화하며(S530), 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화한다(S540).

[71] 단계 S510에서, 움직임 벡터 부호화기(110)는 주변 블록의 움직임 정보를 이용하여 카메라의 움직임 여부를 판단하고 판단되는 카메라의 움직임 여부를 기초로 하나의 후보 예측 움직임 벡터 집합을 선택할 수 있다.

[72] 또한, 단계 S510에서, 움직임 벡터 부호화기(110)는 주변 블록의 움직임 정보를 이용하여 주변 블록의 객체 움직임을 판단하고, 판단되는 주변 블록의 객체 움직임을 기초로 하나의 후보 예측 움직임 벡터 집합을 선택할 수 있다. 이를 위해, 움직임 벡터 부호화기(110)는 주변 블록의 예측 움직임 벡터를 이용하여 주변 블록의 객체 움직임을 판단할 수 있다.

[73] 또한, 단계 S510에서, 움직임 벡터 부호화기(110)는 단계 S510에서, 주변 블록의 움직임 정보를 이용하여 주변 블록의 객체 움직임을 판단하고 판단되는 객체 움직임을 기초로 후보 예측 움직임 벡터를 선택하며, 주변 블록의 움직임 정보를 이용하여 카메라의 움직임 여부를 판단하고 판단되는 카메라의 움직임 여부를 기초로 후보 예측 움직임 벡터를 선택하며, 복수 개의 후보 예측 움직임 벡터 집합 중에서 객체 움직임을 기초로 선택되는 후보 예측 움직임 벡터 및 카메라의 움직임 여부를 기초로 선택되는 후보 예측 움직임 벡터를 포함하는

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000156

후보 예측 움직임 벡터 집합을 하나의 후보 예측 움직임 벡터 집합으로서 선택할 수 있다.

[74] 도 6은 본 발명의 일 실시예에 따른 영상 부호화 방법을 설명하기 위한 순서도이다.

[75] 본 발명의 일 실시예에 따른 영상 부호화 방법에 따르면, 영상 부호화 장치(100)는 현재 블록의 현재 움직임 벡터를 결정하고(S610), 결정되는 현재 움직임 벡터를 이용하여 현재 블록을 예측 부호화하며(S620), 복수 개의 후보 예측 움직임 벡터 집합 중 현재 블록의 주변 블록의 움직임 정보를 기초로 선택되는 후보 예측 움직임 벡터 집합에서 예측 움직임 벡터를 선택하며(S630), 현재 움직임 벡터와 선택되는 예측 움직임 벡터의 차이인 차분 벡터와 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화한다(S640).

[76] 도 7은 본 발명의 일 실시예에 따른 영상 복호화 장치를 간략하게 나타낸 블록 구성도이다.

[77] 본 발명의 일 실시예에 따른 영상 복호화 장치(700)는 움직임 벡터 복호화기(Motion Vector Decoder, 710) 및 영상 복호화기(Video Decoder, 720)를 포함하여 구성될 수 있다.

[78] 움직임 벡터 복호화기(710)는 비트스트림으로부터 추출되는 움직임 벡터 데이터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원하고, 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하며, 선택되는 후보 예측 움직임 벡터 집합에서 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하며, 복원되는 차분 벡터와 선택되는 예측 움직임 벡터를 가산하여 현재 블록의 현재 움직임 벡터를 복원한다. 움직임 벡터 복호화기(710)에 대해서는 후술하는 과정에서 도 8을 통해 상세히 설명한다.

[79] 영상 복호화기(720)는 비트스트림으로부터 추출되는 영상 데이터를 복원되는 현재 움직임 벡터를 이용하여 예측 복호화하여 현재 블록을 복원한다. 이를 위해, 영상 복호화기(720)는 복호화기(Decoder), 역 양자화기 및 역 변환기, 예측기, 가산기, 디블로킹 필터, 메모리 등을 포함하여 구성될 수 있다. 여기서, 복호화기는 비트스트림으로부터 추출되는 영상 데이터를 복호화하여 양자화된 변환 계수를 복원하고, 역 양자화기 및 역 변환기는 복원되는 양자화된 변환 계수를 역 양자화 및 역 변환하여 잔여 블록을 복원하며, 예측기는 움직임 벡터 복호화기(710)에 의해 복원되는 현재 블록의 현재 움직임 벡터를 이용하여 현재 블록의 움직임을 보상함으로써 예측 블록을 생성하며, 가산기는 복원되는 잔여 블록과 예측 블록을 가산하여 현재 블록을 복원할 수 있다. 복원되는 현재 블록은 디블로킹 필터에 의해 디블로킹 필터링되고 픽처 단위로 누적되어 복원 영상으로서 출력되거나 메모리에 저장되어 예측기가 다음 블록 또는 다음 픽처를 예측하는데 활용된다.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000157

[80] 도 8은 본 발명의 일 실시예에 따른 움직임 벡터 복호화 장치를 간략하게 나타낸 블록 구성도이다.

[81] 본 발명의 일 실시예에 따른 움직임 벡터 복호화 장치는 도 7을 통해 전술한 영상 복호화 장치(700)에서는 움직임 벡터 복호화기(710)로 구현될 수 있다. 이하에서는 설명의 편의를 위해, 본 발명의 일 실시예에 따른 움직임 벡터 복호화 장치를 움직임 벡터 복호화기(710)라 칭한다.

[82] 움직임 벡터 복호화기(710)는 후보 예측 움직임 벡터 집합 선택기(810), 예측 움직임 벡터 복원기(Predicted Motion Vector Reconstructer, 820) 및 현재 움직임 벡터 복원기(Current Motion Vector Reconstructer, 830)를 포함하여 구성될 수 있다.

[83] 후보 예측 움직임 벡터 집합 선택기(810)는 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택한다. 여기서, 후보 예측 움직임 벡터 집합 선택기(810)는 도 2를 통해 전술한 후보 예측 움직임 벡터 집합 선택기(210)와 동일 또는 유사하므로 상세한 설명은 생략한다.

[84] 예측 움직임 벡터 복원기(820)는 움직임 벡터 데이터로부터 추출되는 인덱스 데이터를 복호화하여 예측 움직임 벡터 인덱스를 복원하고, 선택되는 후보 예측 움직임 벡터 집합에서 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 복원한다. 즉, 예측 움직임 벡터 복원기(820)는 움직임 벡터 데이터로부터 인덱스 데이터를 추출하고 복호화하여 예측 움직임 벡터 인덱스를 복원하고, 후보 예측 움직임 벡터 집합 선택기(810)에 의해 선택되는 후보 예측 움직임 벡터 집합 내의 후보 예측 움직임 벡터들 중에서 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 복원한다.

[85] 예를 들어, 후보 예측 움직임 벡터 집합 선택기(810)에 의해 선택되는 후보 예측 움직임 벡터 집합이 $\{MV_a, MV_{H.264}\}$이고, MV_a의 예측 움직임 벡터 인덱스는 '0'이고 $MV_{H.264}$의 예측 움직임 벡터 인덱스는 '1'로 설정되어 있으며, 움직임 벡터 데이터로부터 추출되는 인덱스 데이터가 '0'의 비트라고 가정하면, 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터는 MV_a가 되어, MV_a가 예측 움직임 벡터로서 복원될 수 있다. 여기서, 후보 예측 움직임 벡터 집합별로 각 집합 내에 포함되는 후보 예측 움직임 벡터에 대한 예측 움직임 벡터 인덱스는 영상 부호화 장치(100)와 영상 복호화 장치(700) 또는 움직임 벡터 부호화기(110)와 움직임 벡터 복호화기(710)에 미리 동일하게 설정되어 저장되어야 한다.

[86] 현재 움직임 벡터 복원기(830)는 움직임 벡터 데이터로부터 추출되는 차분 벡터 데이터를 복호화하여 차분 벡터를 복원하고, 복원되는 차분 벡터와 복원되는 예측 움직임 벡터를 가산하여 현재 블록의 현재 움직임 벡터를 복원한다. 즉, 현재 움직임 벡터 복원기(830)는 움직임 벡터 데이터로부터 차분

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000158

벡터 데이터를 추출하고 복호화하여 차분 벡터를 복원하고, 복원되는 차분 벡터와 예측 움직임 벡터 복원기(820)에 의해 복원되는 예측 움직임 벡터를 가산하여 현재 움직임 벡터를 복원한다.

[87] 도 9는 본 발명의 일 실시예에 따른 움직임 벡터 복호화 방법을 설명하기 위한 순서도이다.

[88] 본 발명의 일 실시예에 따른 움직임 벡터 복호화 방법에 따르면, 움직임 벡터 복호화기(710)는 움직임 벡터 데이터로부터 추출되는 차분 벡터 데이터와 인덱스 데이터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원하고(S910), 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하며(S920), 선택되는 후보 예측 움직임 벡터 집합에서 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 복원하며(S930), 복원되는 차분 벡터와 복원되는 예측 움직임 벡터를 가산하여 현재 블록의 현재 움직임 벡터를 복원한다(S940).

[89] 단계 S920에서, 움직임 벡터 복호화기(710)는 주변 블록의 움직임 정보를 이용하여 카메라의 움직임 여부를 판단하고 판단되는 카메라의 움직임 여부를 기초로 하나의 후보 예측 움직임 벡터 집합을 선택할 수 있다.

[90] 또한, 단계 S920에서, 움직임 벡터 복호화기(710)는 주변 블록의 움직임 정보를 이용하여 주변 블록의 객체 움직임을 판단하고, 판단되는 주변 블록의 객체 움직임을 기초로 하나의 후보 예측 움직임 벡터 집합을 선택할 수 있다. 이를 위해, 움직임 벡터 복호화기(710)는 주변 블록의 예측 움직임 벡터를 이용하여 주변 블록의 객체 움직임을 판단할 수 있다.

[91] 또한, 단계 S920에서, 움직임 벡터 복호화기(710)는 주변 블록의 움직임 정보를 이용하여 주변 블록의 객체 움직임을 판단하고 판단되는 객체 움직임을 기초로 후보 예측 움직임 벡터를 선택하며, 주변 블록의 움직임 정보를 이용하여 카메라의 움직임 여부를 판단하고 판단되는 카메라의 움직임 여부를 기초로 후보 예측 움직임 벡터를 선택하며, 복수 개의 후보 예측 움직임 벡터 집합 중에서 객체 움직임을 기초로 선택되는 후보 예측 움직임 벡터 및 카메라의 움직임 여부를 기초로 선택되는 후보 예측 움직임 벡터를 포함하는 후보 예측 움직임 벡터 집합을 하나의 후보 예측 움직임 벡터 집합으로서 선택할 수 있다.

[92] 도 10은 본 발명의 일 실시예에 따른 영상 복호화 방법을 설명하기 위한 순서도이다.

[93] 본 발명의 일 실시예에 따른 영상 복호화 방법에 따르면, 영상 복호화 장치(700)는 비트스트림으로부터 추출되는 움직임 벡터 데이터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원하고(S1010), 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하며(S1020), 선택되는 후보 예측 움직임 벡터 집합에서 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000159

예측 움직임 벡터를 예측 움직임 벡터로서 선택하며(S1030), 복원되는 차분 벡터와 선택되는 예측 움직임 벡터를 가산하여 현재 블록의 현재 움직임 벡터를 복원하며(S1040), 비트스트림으로부터 추출되는 영상 데이터를 복원되는 현재 움직임 벡터를 이용하여 예측 복호화하여 현재 블록을 복원할 수 있다.

[94] 한편, 이상에서는 영상과 움직임 벡터가 블록 단위로 예측 부호화되고 복호화되는 것으로 예를 들어 설명했지만, 영상과 움직임 벡터가 반드시 블록 단위로 예측 부호화되고 복호화되어야 하는 것은 아니다. 예를 들어, 블록 단위가 아닌, 슬라이스, 픽처, 시퀀스 단위와 같은 소정의 부호화 단위로 예측 부호화될 수도 있고 블록의 형태가 아닌 비정형의 다양한 영역의 형태로 예측 부호화될 수도 있다.

[95] 이상에서 전술한 바와 같이, 본 발명의 일 실시예에 따르면, 이미 부호화되고 복호화되어 복원된 주변 블록의 움직임 정보를 이용하여 블록 단위, 슬라이스 단위, 픽처 단위, 시퀀스 단위 등 소정의 부호화 단위마다 적응적으로 해당 부호화 단위에 적합한 후보 예측 움직임 벡터 집합을 선택함으로써, 부호화하고자 하는 움직임 벡터와 더욱 유사한 예측 움직임 벡터를 선택하여 차분 벡터의 크기를 줄이면서도 선택한 후보 예측 움직임 벡터 집합에 대한 정보를 부호화할 필요가 없으므로, 움직임 벡터를 부호화하는 데 소요되는 비트량을 줄일 수 있으며 그에 따라 영상의 압축 효율을 향상시킬 수 있다.

[96] 이상에서, 본 발명의 실시예를 구성하는 모든 구성 요소들이 하나로 결합하거나 결합하여 동작하는 것으로 설명되었다고 해서, 본 발명이 반드시 이러한 실시예에 한정되는 것은 아니다. 즉, 본 발명의 목적 범위 안에서라면, 그 모든 구성 요소들이 하나 이상으로 선택적으로 결합하여 동작할 수도 있다. 또한, 그 모든 구성 요소들이 각각 하나의 독립적인 하드웨어로 구현될 수 있지만, 각 구성 요소들의 그 일부 또는 전부가 선택적으로 조합되어 하나 또는 복수 개의 하드웨어에서 조합된 일부 또는 전부의 기능을 수행하는 프로그램 모듈을 갖는 컴퓨터 프로그램으로서 구현될 수도 있다. 그 컴퓨터 프로그램을 구성하는 코드들 및 코드 세그먼트들은 본 발명의 기술 분야의 당업자에 의해 용이하게 추론될 수 있을 것이다. 이러한 컴퓨터 프로그램은 컴퓨터가 읽을 수 있는 저장매체(Computer Readable Media)에 저장되어 컴퓨터에 의하여 읽혀지고 실행됨으로써, 본 발명의 실시예를 구현할 수 있다. 컴퓨터 프로그램의 저장매체로서는 자기 기록매체, 광 기록매체, 캐리어 웨이브 매체 등이 포함될 수 있다.

[97] 또한, 이상에서 기재된 "포함하다", "구성하다" 또는 "가지다" 등의 용어는, 특별히 반대되는 기재가 없는 한, 해당 구성 요소가 내재할 수 있음을 의미하는 것이므로, 다른 구성 요소를 제외하는 것이 아니라 다른 구성 요소를 더 포함할 수 있는 것으로 해석되어야 한다. 기술적이거나 과학적인 용어를 포함한 모든 용어들은, 다르게 정의되지 않는 한, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미가 있다. 사전에

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000160

정의된 용어와 같이 일반적으로 사용되는 용어들은 관련 기술의 문맥상의 의미와 일치하는 것으로 해석되어야 하며, 본 발명에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[98] 이상의 설명은 본 발명의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명에 개시된 실시예들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시예에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

산업상 이용가능성

[99] 이상에서 설명한 바와 같이 본 발명은 영상을 부호화하고 복호화하는 영상 압축 처리 분야에 적용되어, 효율적인 후보 예측 움직임 벡터 집합을 선택하여 움직임 벡터를 부호화함에 따라 부호화할 차분 벡터의 크기를 줄이면서도 어떠한 후보 예측 움직임 벡터 집합이 선택되었는지를 나타내기 위한 부가 정보를 부호화하지 않아도 되므로, 움직임 벡터의 압축 효율을 향상시켜 결과적으로 영상의 압축 효율을 향상시킬 수 있는 효과를 발생하는 매우 유용한 발명이다.

[100]

[101] CROSS-REFERENCE TO RELATED APPLICATION

[102] 본 특허출원은 2009년 11월 18일 한국에 출원한 특허출원번호 제 10-2009-0111302 호에 대해 미국 특허법 119(a)조(35 U.S.C § 119(a))에 따라 우선권을 주장하면, 그 모든 내용은 참고문헌으로 본 특허출원에 병합된다. 아울러, 본 특허출원은 미국 이외에 국가에 대해서도 위와 동일한 동일한 이유로 우선권을 주장하면 그 모든 내용은 참고문헌으로 본 특허출원에 병합된다.

청구범위

[청구항 1] 움직임 벡터를 부호화하는 장치에 있어서,
현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하는 후보 예측 움직임 벡터 집합 선택기;
상기 선택되는 후보 예측 움직임 벡터 집합 내의 후보 예측 움직임 벡터 중에서 하나의 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하는 예측 움직임 벡터 선택기;
상기 현재 블록의 움직임 벡터인 현재 움직임 벡터와 상기 선택되는 예측 움직임 벡터의 차이인 차분 벡터를 부호화하는 차분 벡터 부호화기; 및
상기 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화하는 예측 움직임 벡터 부호화기
를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 장치.

[청구항 2] 영상을 부호화하는 장치에 있어서,
현재 블록의 현재 움직임 벡터를 결정하고, 상기 결정되는 현재 움직임 벡터를 이용하여 상기 현재 블록을 예측 부호화하는 영상 부호화기; 및
복수 개의 후보 예측 움직임 벡터 집합 중 상기 현재 블록의 주변 블록의 움직임 정보를 기초로 선택되는 후보 예측 움직임 벡터 집합에서 예측 움직임 벡터를 선택하고, 현재 움직임 벡터와 상기 선택되는 예측 움직임 벡터의 차이인 차분 벡터 및 상기 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화하는 움직임 벡터 부호화기
를 포함하는 것을 특징으로 하는 영상 부호화 장치.

[청구항 3] 움직임 벡터를 복호화하는 장치에 있어서,
현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하는 후보 예측 움직임 벡터 집합 선택기;
움직임 벡터 데이터로부터 추출되는 인덱스 데이터를 복호화하여 예측 움직임 벡터 인덱스를 복원하고, 상기 선택되는 후보 예측 움직임 벡터 집합에서 상기 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 복원하는 예측 움직임 벡터 복원기; 및
상기 움직임 벡터 데이터로부터 추출되는 차분 벡터 데이터를 복호화하여 차분 벡터를 복원하고, 상기 복원되는 차분 벡터와 상기 복원되는 예측 움직임 벡터를 가산하여 상기 현재 블록의

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000162

현재 움직임 벡터를 복원하는 현재 움직임 벡터 복원기
를 포함하는 것을 특징으로 하는 움직임 벡터 복호화 장치.

[청구항 4] 영상을 복호화하는 장치에 있어서,
비트스트림으로부터 추출되는 움직임 벡터 데이터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원하고, 현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하며, 상기 선택되는 후보 예측 움직임 벡터 집합에서 상기 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하며, 상기 복원되는 차분 벡터와 상기 선택되는 예측 움직임 벡터를 가산하여 상기 현재 블록의 현재 움직임 벡터를 복원하는 움직임 벡터 복호화기; 및
상기 비트스트림으로부터 추출되는 영상 데이터를 상기 복원되는 현재 움직임 벡터를 이용하여 예측 복호화하여 상기 현재 블록을 복원하는 영상 복호화기
를 포함하는 것을 특징으로 하는 영상 복호화 장치.

[청구항 5] 움직임 벡터를 부호화하는 방법에 있어서,
현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계;
상기 선택되는 후보 예측 움직임 벡터 집합 내의 후보 예측 움직임 벡터 중에서 하나의 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하는 단계;
상기 현재 블록의 움직임 벡터인 현재 움직임 벡터와 상기 선택되는 예측 움직임 벡터의 차이인 차분 벡터를 부호화하는 단계; 및
상기 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화하는 단계
를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 방법.

[청구항 6] 제 5 항에 있어서,
상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계는,
상기 주변 블록의 움직임 정보를 이용하여 카메라의 움직임 여부를 판단하고 상기 판단되는 카메라의 움직임 여부를 기초로 상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 것을 특징으로 하는 움직임 벡터 부호화 방법.

[청구항 7] 제 5 항에 있어서,
상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계는,
상기 주변 블록의 움직임 정보를 이용하여 상기 주변 블록의 객체

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000163

움직임을 판단하고, 상기 판단되는 주변 블록의 객체 움직임을 기초로 상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 것을 특징으로 하는 움직임 벡터 부호화 방법.

[청구항 8] 제 7 항에 있어서,
상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계는,
상기 주변 블록의 예측 움직임 벡터를 이용하여 상기 주변 블록의 객체 움직임을 판단하는 것을 특징으로 하는 움직임 벡터 부호화 방법.

[청구항 9] 제 5 항에 있어서,
상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계는,
상기 주변 블록의 움직임 정보를 이용하여 상기 주변 블록의 객체 움직임을 판단하고 상기 판단되는 객체 움직임을 기초로 후보 예측 움직임 벡터를 선택하는 단계;
상기 주변 블록의 움직임 정보를 이용하여 카메라의 움직임 여부를 판단하고 상기 판단되는 카메라의 움직임 여부를 기초로 후보 예측 움직임 벡터를 선택하는 단계; 및
상기 복수 개의 후보 예측 움직임 벡터 집합 중에서 상기 객체 움직임을 기초로 선택되는 후보 예측 움직임 벡터 및 상기 카메라의 움직임 여부를 기초로 선택되는 후보 예측 움직임 벡터를 포함하는 후보 예측 움직임 벡터 집합을 상기 하나의 후보 예측 움직임 벡터 집합으로서 선택하는 단계
를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 방법.

[청구항 10] 영상을 부호화하는 방법에 있어서,
현재 블록의 현재 움직임 벡터를 결정하는 단계;
상기 결정되는 현재 움직임 벡터를 이용하여 상기 현재 블록을 예측 부호화하는 단계;
복수 개의 후보 예측 움직임 벡터 집합 중 상기 현재 블록의 주변 블록의 움직임 정보를 기초로 선택되는 후보 예측 움직임 벡터 집합에서 예측 움직임 벡터를 선택하는 단계; 및
상기 현재 움직임 벡터와 상기 선택되는 예측 움직임 벡터의 차이인 차분 벡터와 상기 선택되는 예측 움직임 벡터를 나타내는 예측 움직임 벡터 인덱스를 부호화하는 단계
를 포함하는 것을 특징으로 하는 영상 부호화 방법.

[청구항 11] 움직임 벡터를 복호화하는 방법에 있어서,
움직임 벡터 데이터로부터 추출되는 차분 벡터 데이터와 인덱스 데이터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원하는 단계;
현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000164

예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계;
상기 선택되는 후보 예측 움직임 벡터 집합에서 상기 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 복원하는 단계; 및
상기 복원되는 차분 벡터와 상기 복원되는 예측 움직임 벡터를 가산하여 상기 현재 블록의 현재 움직임 벡터를 복원하는 단계를 포함하는 것을 특징으로 하는 움직임 벡터 복호화 방법.

[청구항 12] 제 11 항에 있어서,
상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계는,
상기 주변 블록의 움직임 정보를 이용하여 카메라의 움직임 여부를 판단하고 상기 판단되는 카메라의 움직임 여부를 기초로 상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 것을 특징으로 하는 움직임 벡터 복호화 방법.

[청구항 13] 제 11 항에 있어서,
상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계는,
상기 주변 블록의 움직임 정보를 이용하여 상기 주변 블록의 객체 움직임을 판단하고, 상기 판단되는 주변 블록의 객체 움직임을 기초로 상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 것을 특징으로 하는 움직임 벡터 복호화 방법.

[청구항 14] 제 13 항에 있어서,
상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계는,
상기 주변 블록의 예측 움직임 벡터를 이용하여 상기 주변 블록의 객체 움직임을 판단하는 것을 특징으로 하는 움직임 벡터 복호화 방법.

[청구항 15] 제 11 항에 있어서,
상기 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계는,
상기 주변 블록의 움직임 정보를 이용하여 상기 주변 블록의 객체 움직임을 판단하고 상기 판단되는 객체 움직임을 기초로 후보 예측 움직임 벡터를 선택하는 단계;
상기 주변 블록의 움직임 정보를 이용하여 카메라의 움직임 여부를 판단하고 상기 판단되는 카메라의 움직임 여부를 기초로 후보 예측 움직임 벡터를 선택하는 단계; 및
상기 복수 개의 후보 예측 움직임 벡터 집합 중에서 상기 객체 움직임을 기초로 선택되는 후보 예측 움직임 벡터 및 상기 카메라의 움직임 여부를 기초로 선택되는 후보 예측 움직임 벡터를 포함하는 후보 예측 움직임 벡터 집합을 상기 하나의 후보 예측 움직임 벡터 집합으로서 선택하는 단계

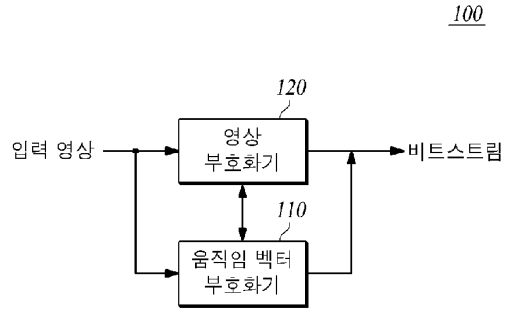
Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000165

를 포함하는 것을 특징으로 하는 움직임 벡터 복호화 방법.

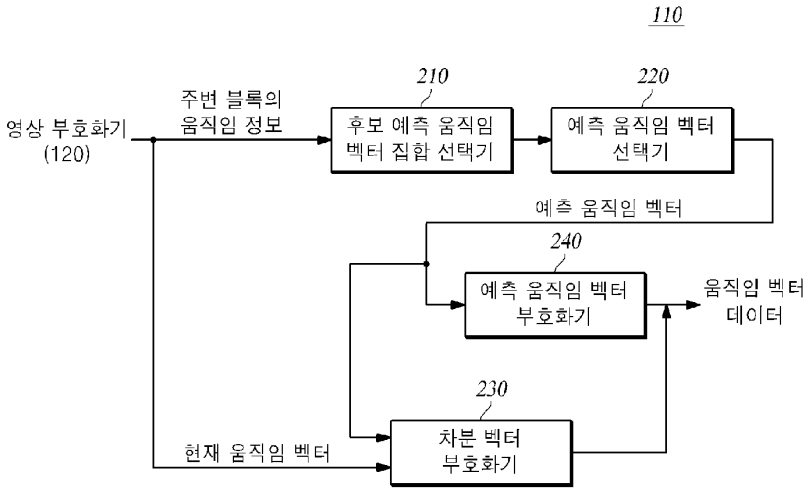
[청구항 16] 영상을 복호화하는 방법에 있어서,
비트스트림으로부터 추출되는 움직임 벡터 데이터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원하는 단계;
현재 블록의 주변 블록의 움직임 정보를 이용하여 복수 개의 후보 예측 움직임 벡터 집합 중에서 하나의 후보 예측 움직임 벡터 집합을 선택하는 단계;
상기 선택되는 후보 예측 움직임 벡터 집합에서 상기 복원되는 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 선택하는 단계;
상기 복원되는 차분 벡터와 상기 선택되는 예측 움직임 벡터를 가산하여 상기 현재 블록의 현재 움직임 벡터를 복원하는 단계; 및
상기 비트스트림으로부터 추출되는 영상 데이터를 상기 복원되는 현재 움직임 벡터를 이용하여 예측 복호화하여 상기 현재 블록을 복원하는 단계
를 포함하는 것을 특징으로 하는 영상 복호화 방법.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000166

[Fig. 1]

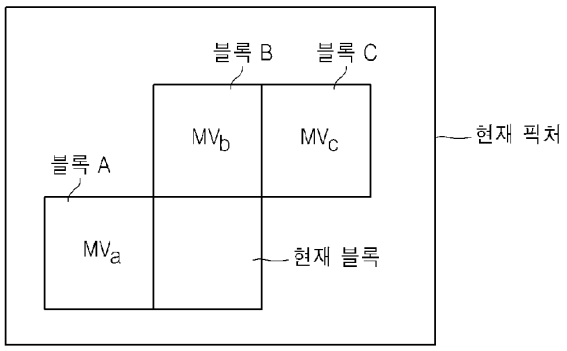

100
120
입력 영상
영상 부호화기
비트스트림
110
움직임 벡터 부호화기

[Fig. 2]

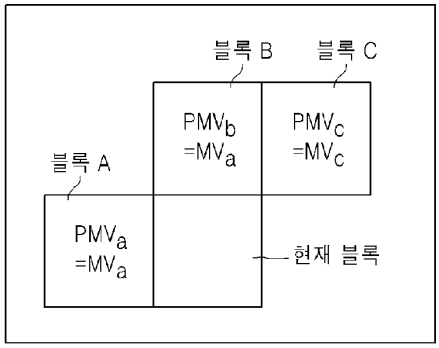

110
영상 부호화기 (120)
주변 블록의 움직임 정보
210
후보 예측 움직임 벡터 집합 선택기
220
예측 움직임 벡터 선택기
예측 움직임 벡터
240
예측 움직임 벡터 부호화기
움직임 벡터 데이터
230
차분 벡터 부호화기
현재 움직임 벡터

[Fig. 3]

3A



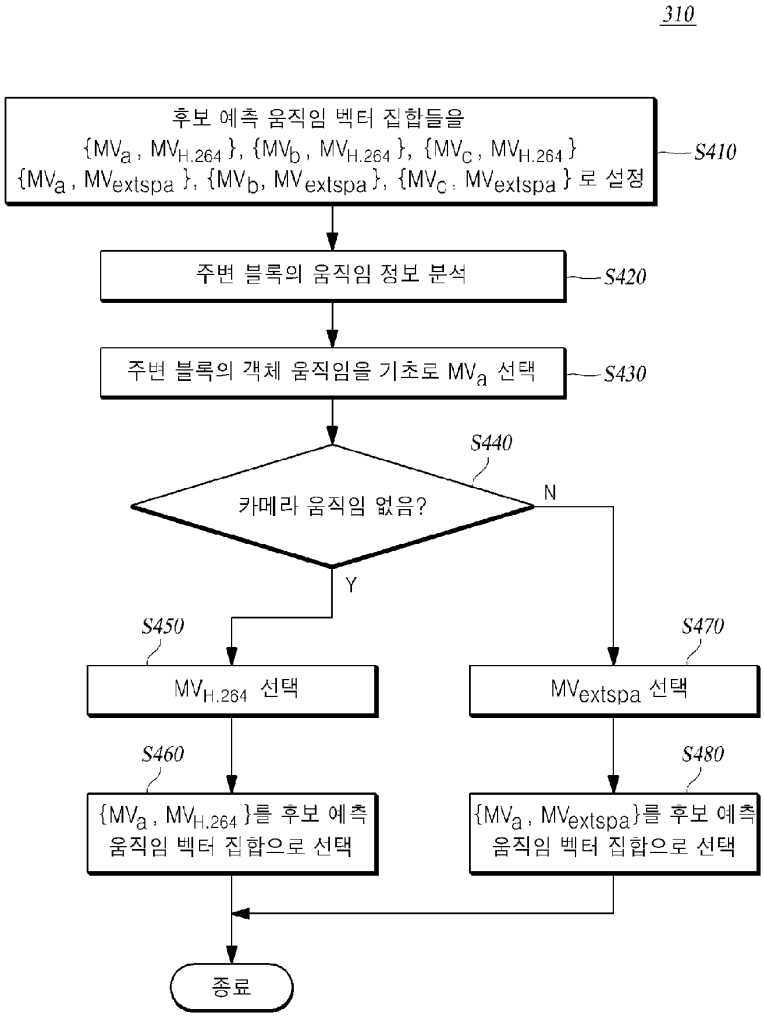
3B



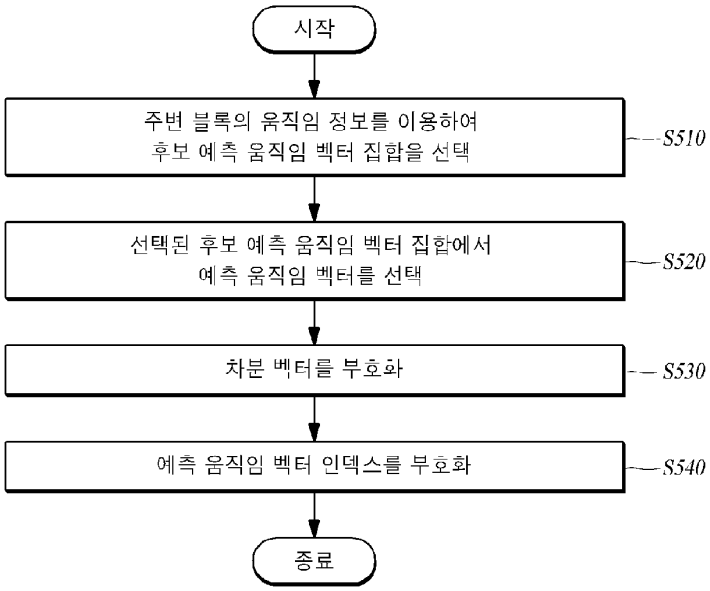
3C

주변 블록의 움직임 정보 = {MV_a , MV_b , MV_c , PMV_a , PMV_b , PMV_c}

[Fig. 4]

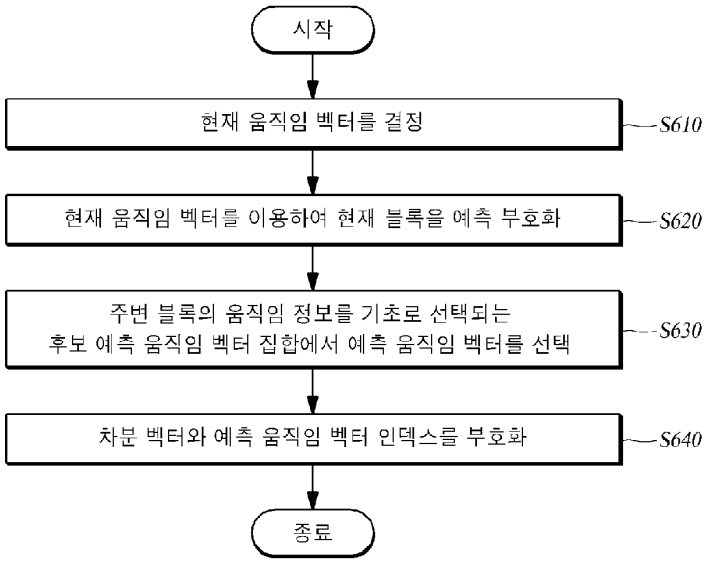


[Fig. 5]

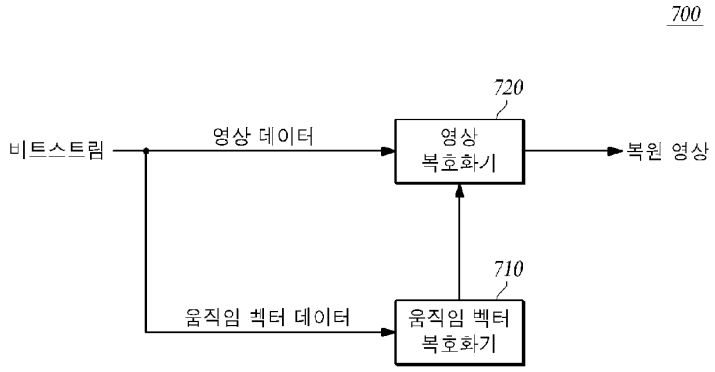


Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000169

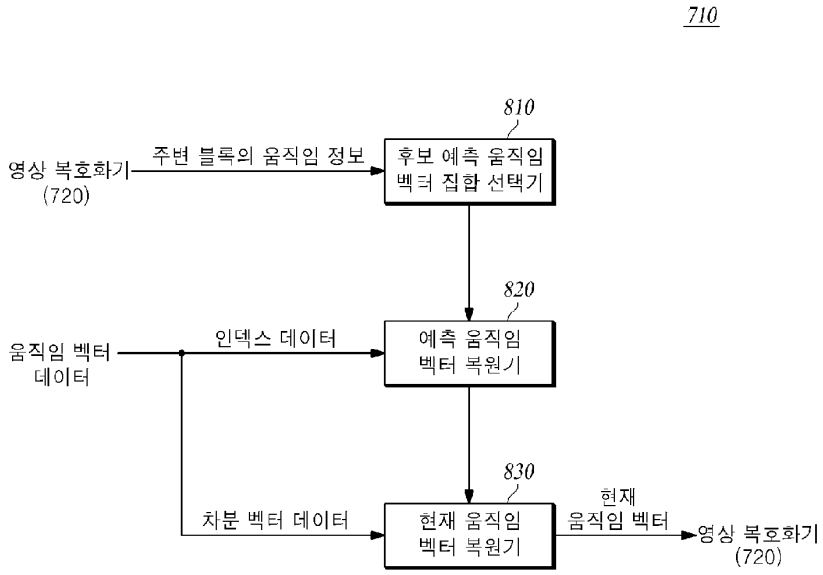
[Fig. 6]



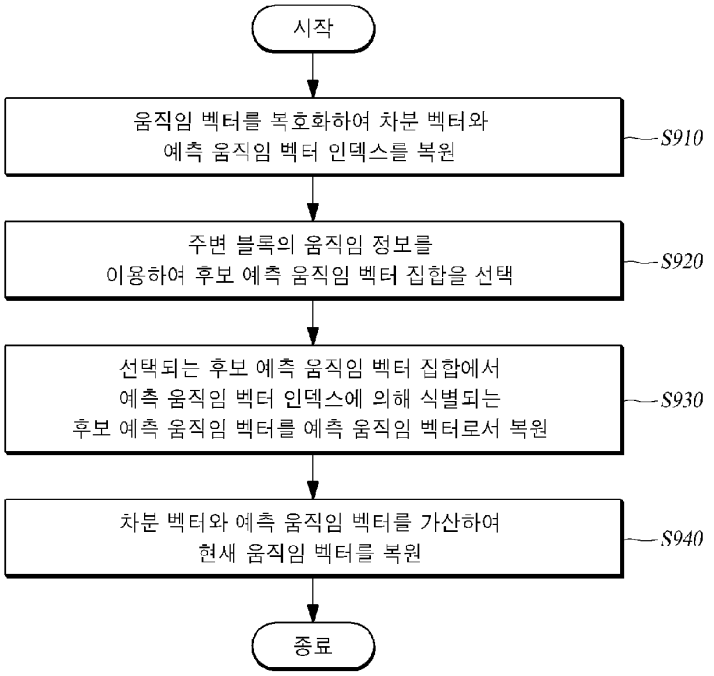
[Fig. 7]



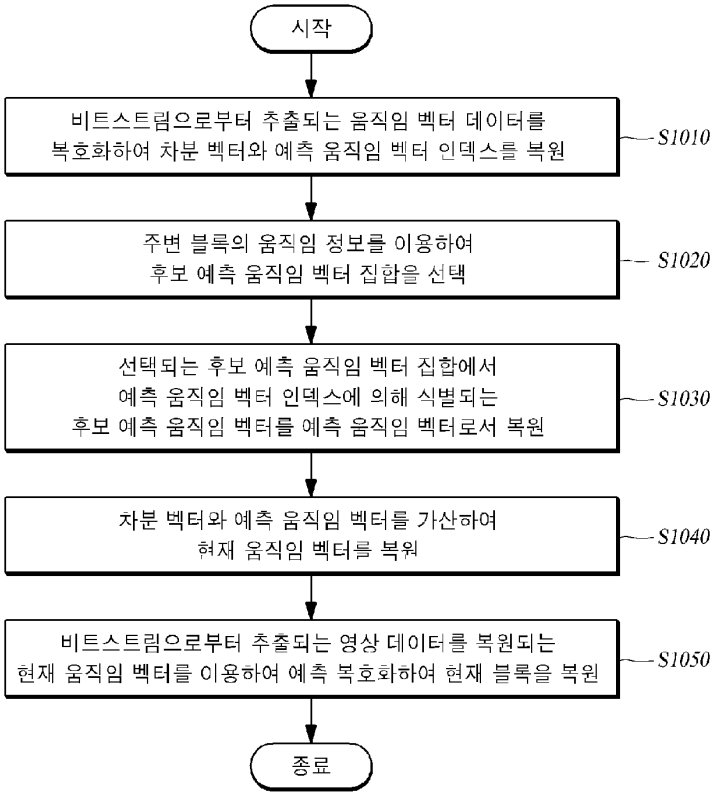
[Fig. 8]



[Fig. 9]


시작
움직임 벡터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원
S910
주변 블록의 움직임 정보를 이용하여 후보 예측 움직임 벡터 집합을 선택
S920
선택되는 후보 예측 움직임 벡터 집합에서 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 복원
S930
차분 벡터와 예측 움직임 벡터를 가산하여 현재 움직임 벡터를 복원
S940
종료

[Fig. 10]


시작
비트스트림으로부터 추출되는 움직임 벡터 데이터를 복호화하여 차분 벡터와 예측 움직임 벡터 인덱스를 복원
S1010
주변 블록의 움직임 정보를 이용하여 후보 예측 움직임 벡터 집합을 선택
S1020
선택되는 후보 예측 움직임 벡터 집합에서 예측 움직임 벡터 인덱스에 의해 식별되는 후보 예측 움직임 벡터를 예측 움직임 벡터로서 복원
S1030
차분 벡터와 예측 움직임 벡터를 가산하여 현재 움직임 벡터를 복원
S1040
비트스트림으로부터 추출되는 영상 데이터를 복원되는 현재 움직임 벡터를 이용하여 예측 복호화하여 현재 블록을 복원
S1050
종료

Electronic Acknowledgement Receipt

EFS ID:	15579218
Application Number:	13666680
International Application Number:	
Confirmation Number:	4782
Title of Invention:	METHOD FOR CODING AND AN APPARATUS
First Named Inventor/Applicant Name:	Mehmet Oguz BICI
Customer Number:	73658
Filer:	Thomas Joseph Arria/thao pham
Filer Authorized By:	Thomas Joseph Arria
Attorney Docket Number:	NC77198US-NP
Receipt Date:	22-APR-2013
Filing Date:	01-NOV-2012
Time Stamp:	15:55:51
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) Form (SB08)	77198_IDS.pdf	612036 589f3026006778cb6803b0090f6d1baa211799ae	no	4

Warnings:	
Information:	

2	Non Patent Literature	77198_ISR_WO.pdf	1918766 092c169f17cdb2416782db54eb72d5f1d2cefb53	no	12
Warnings:					
Information:					
3	Non Patent Literature	OUDIN.PDF	468422 315377bd759752c6fa371689a7c187886a4997a1	no	6
Warnings:					
Information:					
4	Non Patent Literature	SULLIVAN.PDF	1726067 7cc3203335537fbc186f548ac62e2b9ff5ee72f8	no	20
Warnings:					
Information:					
5	Foreign Reference	WO2011062392A2.pdf	3674943 751e69a655a915ff41fd86d434c8b5b122fa388a	no	28
Warnings:					
Information:					
			Total Files Size (in bytes):	8400234	

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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/666,680	11/01/2012	Mehmet Oguz BICI	NC77198US-NP

CONFIRMATION NO. 4782

73658
Nokia, Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086

PUBLICATION NOTICE



Title:METHOD FOR CODING AND AN APPARATUS

Publication No.US-2013-0114723-A1
Publication Date:05/09/2013

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently http://pair.uspto.gov/. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000174

To: brian.odonoghue@nokia.com,thao.pham@nokia.com,sandy.fong-hou@nokia.com
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 73658

May 10, 2013 05:33:24 AM

Dear PAIR Customer:

Nokia, Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 73658 , 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.
13666680	NTC.PUB	05/09/2013	NC77198US-NP

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

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 13/666,680

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	280
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	600
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	720
TOTAL CLAIMS (37 CFR 1.16(i))	20 minus 20 =	*			OR	x 80 =	0.00
INDEPENDENT CLAIMS (37 CFR 1.16(h))	8 minus 3 =	* 5				x 420 =	2100
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 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).						0.00
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0.00
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	3700

APPLICATION AS AMENDED - PART II

		(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($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	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	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
AMENDMENT B	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 found in the appropriate box in column 1.

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	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
13/666,680	11/01/2012	2631	2640	NC77198US-NP	20	8

CONFIRMATION NO. 4782

UPDATED FILING RECEIPT

73658
Nokia, Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086

OC000000062918875

Date Mailed: 08/05/2013

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Mehmet Oguz BICI, Tampere, TURKEY;
Jani LAINEMA, Tampere, FINLAND;
Kemal UGUR, Tampere, FINLAND;

Applicant(s)

Mehmet Oguz BICI, Tampere, TURKEY;
Jani LAINEMA, Tampere, FINLAND;
Kemal UGUR, Tampere, FINLAND;

Non-Applicant Assignee(s)

NOKIA CORPORATION, Espoo, FINLAND

Power of Attorney: None

Domestic Priority data as claimed by applicant

This appln claims benefit of 61/555,703 11/04/2011

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

If Required, Foreign Filing License Granted: 11/20/2012
The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 13/666,680**
Projected Publication Date: Not Applicable
Non-Publication Request: No

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000177

Early Publication Request: No
Title

METHOD FOR CODING AND AN APPARATUS

Preliminary Class

375

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications:

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000178

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

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
13/666,680	11/01/2012	Mehmet Oguz BICI	NC77198US-NP

CONFIRMATION NO. 4782

73658
Nokia, Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086

IMPROPER CFR REQUEST

OC000000062918776

Date Mailed: 08/05/2013

RESPONSE TO REQUEST FOR CORRECTED FILING RECEIPT

Power of Attorney, Claims, Fees, System Limitations, and Miscellaneous

In response to your request for a corrected Filing Receipt, the Office is unable to comply with your request because:

- Any request to correct or update the name of the applicant must include an application data sheet (ADS) in compliance with 37 CFR 1.76 specifying the correct or updated name of the applicant in the applicant information section. Any request to change the applicant after an original applicant has been specified under 37 CFR 1.46(b) must include a new ADS in compliance with 37 CFR 1.76 specifying the applicant in the applicant information section and comply with 37 CFR 3.71 and 3.73. See 37 CFR 1.46(c).

/zretta/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000180

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
13/666,680	11/01/2012	Mehmet Oguz BICI	NC77198US-NP

CONFIRMATION NO. 4782

IMPROPER CPOA LETTER

73658
Nokia, Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086

OC000000062918749

Date Mailed: 08/05/2013

NOTICE REGARDING POWER OF ATTORNEY

This is in response to the power of attorney filed 01/02/2013. The power of attorney in this application is not accepted for the reason(s) listed below:

- The power of attorney has not been accepted because the party who is giving power of attorney has not been identified. Power of attorney may only be signed by the applicant for patent (37 CFR 1.42) or the patent owner. A patent owner who was not the applicant must appoint any power of attorney in compliance with 37 CFR 3.71 and 3.73. See 37 CFR 1.32(b)(4).

/zretta/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000181

To: brian.odonoghue@nokia.com,thao.pham@nokia.com,sandy.fong-hou@nokia.com
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 73658

Aug 05, 2013 05:31:06 AM

Dear PAIR Customer:

Nokia, Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 73658 , 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.
13666680	APP.FILE.REC	08/05/2013	NC77198US-NP
	M327	08/05/2013	NC77198US-NP
	N572	08/05/2013	NC77198US-NP

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

PLUS Search Results for S/N 13666680, Searched Tue May 26 15:06:59 EDT 2015

The Patent Linguistics Utility System (PLUS) is a USPTO automated search system for U.S. Patents from 1971 to the present PLUS is a query-by-example search system which produces a list of patents that are most closely related linguistically to the application searched. This search was prepared by the staff of the Scientific and Technical Information Center, SIRA.

5719627 99
5815602 99
4989089 99
5428403 99
5781249 99
5838391 99
4851906 98
4989087 98
5235419 98
5371549 98
5414469 98
5457481 98
5506622 98
5521642 98
5532746 98
5546129 98
5557341 98
5574663 98
5579050 98
5581308 98
5587741 98
5598216 98
5612743 98
5614954 98
5617144 98
5619281 98
5625417 98
5627591 98
5638129 98
5642166 98
5646867 98
5654761 98
5689306 98
5694487 98
5717463 98
5719630 98
5724369 98
5731851 98
5751362 98
5862261 98
5903672 98
5991447 98
6005980 98
6020925 98
6023298 98
6026195 98
6057884 98
6091777 98
6154519 98
6205177 98

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.
13/666,680	11/01/2012	Mehmet Oguz BICI	NC77198US-NP	4782

73658 7590 07/06/2015

Nokia USA Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086

EXAMINER
HILAIRE, CLIFFORD

ART UNIT	PAPER NUMBER
2488	

NOTIFICATION DATE	DELIVERY MODE
07/06/2015	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):

thao.pham@nokia.com
sandy.fong-hou@nokia.com
jacqueline.linebarger@nokia.com

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000184

Office Action Summary	Application No. 13/666,680	Applicant(s) BICI ET AL.	
	Examiner CLIFFORD HILAIRE	Art Unit 2488	AIA (First Inventor to File) Status No

-- 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* MONTHS 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 *11/01/2012*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

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) *1-20* 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) *1-20* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ 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).

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).

Certified copies:

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)).

** 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) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date *04/22/2013*.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

DETAILED ACTION

Notice of Pre-AIA or AIA Status

1. The present application is being examined under the pre-AIA first to invent provisions.

Claim Rejections - 35 USC § 112

The following is a quotation of 35 U.S.C. 112(b):

(b) CONCLUSION.—The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention.

The following is a quotation of 35 U.S.C. 112 (pre-AIA), second paragraph:

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.

2. Claims 1, 6, 9, 13, 15, 16, 17, 18, 19 and 20 are rejected under 35 U.S.C. 112(b) or 35 U.S.C. 112 (pre-AIA), second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the inventor or a joint inventor, or for pre-AIA the applicant regards as the invention.

3. Claim 6 and 13 recite the limitation "*the maximum number*" in the first limitation of both claims. There is insufficient antecedent basis for this limitation in the claim.

4. Claim 1, 9, 15, 16, 17, 18, 19 and 20 recite the limitation "*the determined subset of spatial motion vector prediction candidates*" in the next to last limitation of each claims *(i.e. "comparing motion information…" for claims 1,9,15,17,18; "compare motion*

information…" for claim 16 and "means for comparing motion information…" for claims 19 and 20). There is insufficient antecedent basis for this limitation in the claim. For examining purposes, it will be assumed the limitation "*the determined subset of spatial motion vector prediction candidates*" is obtain from the limitation above it *(i.e. "determining a subset of spatial motion vector* ***prediction candidates"*** *instead of "determining a subset of spatial motion vector* ***predictions"****).*

Claim Rejections - 35 USC § 101

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.

35 U.S.C. 101 requires that a claimed invention must fall within one of the four eligible categories of invention (i.e. process, machine, manufacture, or composition of matter) *and* must not be directed to subject matter encompassing a judicially recognized exception as interpreted by the courts. MPEP 2106. The four eligible categories of invention include: (1) process which is an act, or a series of acts or steps, (2) machine which is an concrete thing, consisting of parts, or of certain devices and combination of devices, (3) manufacture which is an article produced from raw or prepared materials by giving to these materials new forms, qualities, properties, or combinations, whether by hand labor or by machinery, and (4) composition of matter which is all compositions of two or more substances and all composite articles, whether they be the results of

chemical union, or of mechanical mixture, or whether they be gases, fluids, powders or solids. MPEP 2106(I).

5. Claims 17 and 18 are rejected under 35 U.S.C. 101 as not falling within one of the four statutory categories of invention because the broadest reasonable interpretation of the instant claims in light of the specification encompasses transitory signals. But, transitory signals are not within one of the four statutory categories (i.e. non-statutory subject matter). *See* MPEP 2106(I). However, claims directed toward a non-transitory computer readable medium may qualify as a manufacture and make the claim patent-eligible subject matter. MPEP 2106(I). Therefore, amending the claims to recite a "*non-transitory computer-readable medium*" would resolve this issue.

Claim Rejections - 35 USC § 103

The following is a quotation of pre-AIA 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, 2, 3, 4, 5, 6, 7, 9 10, 11, 12, 14, 15, 16, 17. 18, 19 and 20 rejected under pre-AIA 35 U.S.C. 103 (a) as being unpatentable over Woo-Jin Han et al.

["Improved Video Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools"] in view of Shen-Chuan Tai et al.*["A Multi-Pass True Motion Estimation Scheme With Motion Vector Propagation for Frame Rate Up-Conversion Applications"].*

Regarding claim 1, Han teaches

1. A method comprising *(i.e. a novel video compression scheme based on a highly flexible hierarchy of unit representation which includes three block concepts: coding unit (CU), prediction unit (PU), and transform unit (TU) - Abstract)*:

receiving a block of pixels including a prediction unit *(i.e. Fig. 1 shows all proposed building blocks of the decoder. All the coding tools are extended to support the proposed flexible architecture. Intra prediction is extended to support an arbitrary number of angles rather than the conventional 9-modes defined in H.264/AVC. Fast integer transforms larger than 8 × 8 are developed to support large TU sizes. Edge definition for the deblocking filter is modified according to the CU, PU, and TU concepts. Partition information in quadtreebased adaptive loop filter (QALF) [13] is replaced with CU*

splitting information- page 1710, ¶ 3);

However, Han does not teach explicitly:

determining a set of spatial motion vector prediction candidates for the block of pixels;

the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

In the same field of endeavor, Tai teaches:

determining a set *(i.e.* $CS^i(\vec{X},t)$*)* of spatial motion vector prediction candidates for the block of pixels *(i.e. The MV search in block in the ith pass is performed in accordance with (1). As shown, the search is confined to the vectors defined in the candidate vector set* $CS^i(\vec{X},t)$*), i.e., a full search is not performed- Page 189, Col 2, ¶ 3)*;

the spatial motion vector prediction candidates being provided with motion information *(i.e. The basic principle of the multi-pass ME strategy proposed in*

this study is to utilize the motion information relating to neighboring blocks generated in the previous pass to gradually refine the accuracy of the MV in the current block- Page 189, Col 2, ¶ 3);

selecting a first spatial motion vector prediction *(i.e.* $CS^i(\vec{X},t) = \{\vec{V}|\vec{V} = \overrightarrow{PMV}^i(\vec{X}) + \vec{U}, \vec{U} \in US^i\}$*- equation 2)* candidate *(i.e.*$\overrightarrow{PMV}^i(\vec{X})$*, predictive motion vectors (PMVs)- Page 190, Col 1, ¶ 2)* from the set of spatial motion vector prediction candidates *(i.e.* $\overrightarrow{PMV}^i(\vec{X})$*, is the PMV of block and is chosen from the MVs of the neighboring blocks and is a set of update vectors- Page 190, Col 1, ¶ 2)* as a potential spatial motion vector prediction candidate to be included in a merge list for the *(i.e. current block)* prediction unit *(i.e. Although the MVs of all eight neighboring blocks can theoretically be used as PMV predictors for the current block- Page 190, Col 1, ¶ 3)*;

determining a subset *(i.e.* $CS^i(\vec{X},t) = \{\vec{V}|\vec{V} = \overrightarrow{PMV}^i(\vec{X}) + \vec{U}, \vec{U} \in US^i\}$*- Equation 2)* of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate *(i.e. by selecting one of the* $\overrightarrow{PMV}$ *from MVs of neighboring blocks, a subset of* $CS^i(\vec{X},t)$ *is defined for different linear combination of that PMV and* $\vec{U} \in US^i$*)*;

comparing motion information of the first spatial motion vector prediction candidate *(i.e.* $\overrightarrow{MV}^i(\vec{X},t) = arg_{\vec{V}\in CS^i(\vec{X},t)} \min\left(e(\vec{V},\vec{X},t)\right), 1 \le i \le 12$*; equation 1)* with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates *(i.e. Each pass*

in the proposed ME scheme is performed using one of the PMV predictor patterns (see Fig. 3) and then one MVin the selected PMV pattern which has minimal distortion is selected as the final PMV of this pass- Page 190; Col 1, ¶ 5); if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other *(i.e. all MV's with non-minimal distortion as inferred above)*, excluding the first spatial motion vector prediction candidate from the merge list *(i.e. all MV's with non-minimal distortion as inferred above will not be a final PMV for that pass)*.

It would have been obvious to one with ordinary skill in the art at the time of invention, to modify the teachings of Han with the teachings of Tai to use the geometric relationship proposed in [2] describing the correlation between the current block position and the motion edge to reduce the number of PMV predictors to be evaluated *(Tai- Page 190, Col 1, ¶ 4)*.

Regarding claim 2, Han and Tai teach all the limitatrions of claim 1. Han further teaches:

comprising selecting spatial motion vector prediction candidates from the set of spatial motion vector prediction candidates as the potential spatial motion vector prediction candidate in a predetermined order *(i.e. comprising selecting spatial motion vector prediction candidates from the set of spatial motion vector prediction candidates as the potential spatial motion vector prediction candidate in a predetermined order- page 1712, Col 2, ¶ 4)*.

Regarding claim 3, Han and Tai teach all the limitations of claim 1.

However, Han does not teach:

comprising comparing motion information of the potential spatial motion vector prediction candidate with motion information of at most one other spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates.

In the same field of endeavor, Tai teaches:

comprising comparing motion information of the potential spatial motion vector prediction candidate *(i.e.* $\overrightarrow{MV}^i(\overrightarrow{X},t) = arg_{\vec{V}\in CS^i(\vec{X},t)} \min\left(e(\vec{V},\vec{X},t)\right), 1 \le i \le 12$*; equation 1)*with motion information of at most one other spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates *(i.e. Each pass in the proposed ME scheme is performed using one of the PMV predictor patterns (see Fig. 3) and then one MVin the selected PMV pattern which has minimal distortion is selected as the final PMV of this pass- Page 190; Col 1, ¶ 5).*

Regarding claim 4, Han and Tai teach all the limitations of claim 1. Han further teaches:

comprising examining whether the received block of pixels is divided into a first prediction unit and a second prediction unit *(i.e. tu size flag is equal to one)*; and if so, excluding the potential spatial motion vector prediction candidate from the

merge list *(i.e. the transform may be applied to the residue generated by multiple PU partitions with different motion vectors or prediction methods)* if the prediction unit is the second prediction unit *(i.e. When the tu size flag is equal to zero, the TU size is set equal to that of the CU which it belongs to. When tu size flag is equal to one, the TU size is set as N × N for symmetric PU splitting's and N/2 × N/2 for asymmetric PU splitting's, respectively. This ensures that the transform which is not applied across motion boundaries can be tested in the rate-distortion optimization process for asymmetric PU partitions. It should be noted that the transform may be applied to the residue generated by multiple PU partitions with different motion vectors or prediction methods- Page 1711, ¶ 4-5)*

Regarding claim 5, Han and Tai teach all the limitations of claim 1. Han further teaches:

further comprising determining a maximum number *(i.e. N)* of spatial motion vector prediction candidates to be included in a merge *(i.e. A')* list *(i.e. Fig. 3 shows the spatially adjacent motion vectors which can be considered the candidate for the prediction. Let A =$\{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_N\}$, A'⊂ A, and $I_{A'}$ denote the set of motion vectors above the current PU, the set of the available motion vectors in A, and the set which includes the indices of the elements in A' respectively- page 1712, col 2, ¶ 3)*; and

limiting the number *(i.e. A'⊂ A)* of spatial motion vector prediction candidates in the merge list smaller or equal to the maximum number *(i.e. The set of the*

available motion vectors, A', is derived from A by verifying whether the corresponding reference index is the same as that of the current PU partition- page 1712, col 2, ¶ 3).

Regarding claim 7, Han and Tai teach all the limitations of claim 1. Han further teaches:

further comprising including a temporal motion prediction candidate into the merge list *(i.e. This paper proposes the AMVP method, which is adapted to the proposed flexible unit representation. It allows the selection of the best predictor from the set which consists of three spatially adjacent motion vectors, their median, and a temporal motion vector- Page 1712, Col 2, ¶ 2)*.

Regarding claim 8, Han and Tai teach all the limitations of claim 1. Han further teaches:

comprising selecting one motion vector prediction candidate *(i.e. the best predictor is selected)* from the merge list to represent a motion vector prediction for the block of pixels*(i.e. the best predictor is selected from a given set through rate-distortion optimization- Page 1712, Col 2, ¶ 1)*.

Regarding claim 9, Han teaches

1. A method comprising *(i.e. a novel video compression scheme based on a highly flexible hierarchy of unit representation which includes three block*

concepts: coding unit (CU), prediction unit (PU), and transform unit (TU)- Abstract):

receiving a encoded block of pixels including a prediction unit *(i.e. Fig. 1 shows all proposed building blocks of the decoder. All the coding tools are extended to support the proposed flexible architecture. Intra prediction is extended to support an arbitrary number of angles rather than the conventional 9-modes defined in H.264/AVC. Fast integer transforms larger than 8 × 8 are developed to support large TU sizes. Edge definition for the deblocking filter is modified according to the CU, PU, and TU concepts. Partition information in quadtreebased adaptive loop filter (QALF) [13] is replaced with CU splitting information- page 1710, ¶ 3)*;

However, Lin does not teach explicitly:

determining a set of spatial motion vector prediction candidates for the encoded block of pixels;

the spatial motion vector prediction candidates being provided with motion information;

selecting a first spatial motion vector prediction candidate from the set of spatial motion vector prediction candidates as a potential spatial motion vector prediction candidate to be included in a merge list for the prediction unit;

determining a subset of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate;

comparing motion information of the first spatial motion vector prediction candidate with motion information of the spatial motion vector prediction

candidate in the determined subset of spatial motion vector prediction candidates;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other, excluding the first spatial motion vector prediction candidate from the merge list.

In the same field of endeavor, Tai teaches:

determining a set *(i.e.* $CS^i(\vec{X},t)$*)* of spatial motion vector prediction candidates for the encoded block of pixels *(i.e. The MV search in block in the ith pass is performed in accordance with (1). As shown, the search is confined to the vectors defined in the candidate vector set* $CS^i(\vec{X},t)$*), i.e., a full search is not performed- Page 189, Col 2, ¶ 3)*;

the spatial motion vector prediction candidates being provided with motion information *(i.e. The basic principle of the multi-pass ME strategy proposed in this study is to utilize the motion information relating to neighboring blocks generated in the previous pass to gradually refine the accuracy of the MV in the current block- Page 189, Col 2, ¶ 3)*;

selecting a first spatial motion vector prediction *(i.e.* $CS^i(\vec{X},t) = \{\vec{V}|\vec{V} = \overrightarrow{PMV}^i(\vec{X}) + \vec{U},\vec{U} \in US^i\}$*- equation 2)* candidate *(i.e.*$\overrightarrow{PMV}^i(\vec{X})$*, predictive motion vectors (PMVs)- Page 190, Col 1, ¶ 2)* from the set of spatial motion vector prediction candidates *(i.e.* $\overrightarrow{PMV}^i(\vec{X})$*, is the PMV of block and is chosen from the MVs of the neighboring blocks and is a set of update vectors- Page 190, Col 1, ¶ 2)* as a potential spatial motion vector prediction candidate to be included in a

merge list for the *(i.e. current block)* prediction unit *(i.e. Although the MVs of all eight neighboring blocks can theoretically be used as PMV predictors for the current block- Page 190, Col 1, ¶ 3)*;

determining a subset *(i.e.* $CS^i(\vec{X},t) = \{\vec{V}|\vec{V} = \overrightarrow{PMV}^i(\vec{X}) + \vec{U}, \vec{U} \in US^i\}$*- equation 2)* of spatial motion vector predictions based on the location of the block associated with the first spatial motion vector prediction candidate *(i.e. by selecting one of the* $\overrightarrow{PMV}$ *from MVs of neighboring blocks, a subset of* $CS^i(\vec{X},t)$ *is defined for different linear combination of that PMV and* $\vec{U} \in US^i$*)*;

comparing motion information of the first spatial motion vector prediction candidate *(i.e.* $\overrightarrow{MV}^i(\vec{X},t) = arg_{\vec{V}\in CS^i(\vec{X},t)} \min\left(e(\vec{V},\vec{X},t)\right), 1 \le i \le 12$*; equation 1)* with motion information of the spatial motion vector prediction candidate in the determined subset of spatial motion vector prediction candidates *(i.e. Each pass in the proposed ME scheme is performed using one of the PMV predictor patterns (see Fig. 3) and then one MVin the selected PMV pattern which has minimal distortion is selected as the final PMV of this pass- Page 190; Col 1, ¶ 5)*;

if at least one of the comparisons indicates that the motion vector information of the spatial motion vector prediction candidates correspond with each other *(i.e. all MV's with non-minimal distortion as inferred above)*, excluding the first spatial motion vector prediction candidate from the merge list *(i.e. all MV's with non-minimal distortion as inferred above will not be a final PMV for that pass)*.

Regarding claim 10, Han and Tai teach all the limitations of claim 9.

However, Han does not teach:

comprising comparing motion information of the potential spatial motion vector prediction candidate with motion information of at most one other spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates.

In the same field of endeavor, Tai teaches:

comprising comparing motion information of the potential spatial motion vector prediction candidate *(i.e.* $\overrightarrow{MV}^i(\overrightarrow{X},t) = arg_{\vec{V}\in CS^i(\vec{X},t)} \min\left(e(\vec{V},\vec{X},t)\right), 1 \leq i \leq 12;$ *equation 1)*with motion information of at most one other spatial motion vector prediction candidate of the set of spatial motion vector prediction candidates *(i.e. Each pass in the proposed ME scheme is performed using one of the PMV predictor patterns (see Fig. 3) and then one MVin the selected PMV pattern which has minimal distortion is selected as the final PMV of this pass- Page 190; Col 1, ¶ 5).*

Regarding claim 11, Han and Tai teach all the limitations of claim 9. Han further teaches:

comprising examining whether the received encoded block of pixels is divided into a first prediction unit and a second prediction unit *(i.e. tu size flag is equal to one)*; and if so, excluding the potential spatial motion vector prediction candidate from the merge list *(i.e. the transform may be applied to the residue generated by multiple PU partitions with different motion vectors or prediction methods)* if the

prediction unit is the second prediction unit *(i.e. When the tu size flag is equal to zero, the TU size is set equal to that of the CU which it belongs to. When tu size flag is equal to one, the TU size is set as N × N for symmetric PU splitting's and N/2 × N/2 for asymmetric PU splitting's, respectively. This ensures that the transform which is not applied across motion boundaries can be tested in the rate-distortion optimization process for asymmetric PU partitions. It should be noted that the transform may be applied to the residue generated by multiple PU partitions with different motion vectors or prediction methods- Page 1711, ¶ 4-5)*

Regarding claim 12, Han and Tai teach all the limitations of claim 9. Han further teaches:

further comprising determining a maximum number *(i.e. N)* of spatial motion vector prediction candidates to be included in a merge *(i.e. A')* list *(i.e. Fig. 3 shows the spatially adjacent motion vectors which can be considered the candidate for the prediction. Let* $A = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_N\}$, $A' \subset A$, *and* $I_{A'}$ *denote the set of motion vectors above the current PU, the set of the available motion vectors in A, and the set which includes the indices of the elements in A' respectively- page 1712, col 2, ¶ 3)*; and

limiting the number *(i.e. A'⊂ A)* of spatial motion vector prediction candidates in the merge list smaller or equal to the maximum number *(i.e. The set of the available motion vectors, A', is derived from A by verifying whether the*

corresponding reference index is the same as that of the current PU partition- page 1712, col 2, ¶ 3).

Regarding claim 14, Han and Tai teach all the limitations of claim 9. Han further teaches:

comprising selecting one motion vector prediction candidate *(i.e. the best predictor is selected)* from the merge list to represent a motion vector prediction for the received encoded block of pixels*(i.e. the best predictor is selected from a given set through rate-distortion optimization- Page 1712, Col 2, ¶ 1).*

Regarding claim 15 and 19, apparatus claim 15 and 19 is drawn to the apparatus using/performing the same method as claimed in claim 1. Therefore apparatus claim 15 and 19 corresponds to method claim 1, and is rejected for the same reasons of obviousness as used above.

Regarding claim 16 and 20, apparatus claim 16 and 20 is drawn to the apparatus using/performing the same method as claimed in claim 9. Therefore apparatus claim 16 and 20 corresponds to method claim 9, and is rejected for the same reasons of obviousness as used above.

Regarding claim 17, computer-readable medium storing instructions claim 17 corresponds to the same method as claimed in claim 1, and therefore is also rejected for the same reasons of obviousness as listed above.

Regarding claim 18, computer-readable medium storing instructions claim 50 corresponds to the same method as claimed in claim 9, and therefore is also rejected for the same reasons of obviousness as listed above.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CLIFFORD HILAIRE whose telephone number is (571)272-8397. The examiner can normally be reached on Monday-Friday- 0800-1700, EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Sath V. Perungavoor can be reached on (571)272-7455. 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.

/C. H./
Examiner, Art Unit 2488

/SATH V PERUNGAVOOR/
Supervisory Patent Examiner, Art Unit 2488

Notice of References Cited	Application/Control No. 13/666,680	Applicant(s)/Patent Under Reexamination BICI ET AL.	
	Examiner CLIFFORD HILAIRE	Art Unit 2488	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	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	Han, Woo-Jin, et al. "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools." Circuits and Systems for Video Technology, IEEE Transactions on 20.12 (2010): 1709-1720.
	V	Tai, Shen-Chuan, et al. "A multi-pass true motion estimation scheme with motion vector propagation for frame rate up-conversion applications." Journal of display technology 4.2 (2008): 188-197.
	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. 4782

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
13/666,680	11/01/2012 RULE	375	2488	NC77198US-NP

APPLICANTS

INVENTORS

Mehmet Oguz BICI, Tampere, TURKEY;
Jani LAINEMA, Tampere, FINLAND;
Kemal UGUR, Tampere, FINLAND;

**** CONTINUING DATA ***************************

This appln claims benefit of 61/555,703 11/04/2011

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

11/20/2012

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	/CLIFFORD HILAIRE/ Examiner's Signature	☐ Initials	TURKEY	13	20	8

ADDRESS

Nokia USA Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086
UNITED STATES

TITLE

METHOD FOR CODING AND AN APPARATUS

FILING FEE RECEIVED		
2640	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

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000205

Improved Video Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools

Woo-Jin Han, *Member, IEEE,* Junghye Min, Il-Koo Kim, Elena Alshina, Alexander Alshin, Tammy Lee, Jianle Chen, Vadim Seregin, Sunil Lee, *Member, IEEE,* Yoon Mi Hong, Min-Su Cheon, Nikolay Shlyakhov, Ken McCann, Thomas Davies, and Jeong-Hoon Park

***Abstract*—This paper proposes a novel video compression scheme based on a highly flexible hierarchy of unit representation which includes three block concepts: coding unit (CU), prediction unit (PU), and transform unit (TU). This separation of the block structure into three different concepts allows each to be optimized according to its role; the CU is a macroblock-like unit which supports region splitting in a manner similar to a conventional quadtree, the PU supports nonsquare motion partition shapes for motion compensation, while the TU allows the transform size to be defined independently from the PU. Several other coding tools are extended to arbitrary unit size to maintain consistency with the proposed design, e.g., transform size is extended up to 64×64 and intraprediction is designed to support an arbitrary number of angles for variable block sizes. Other novel techniques such as a new noncascading interpolation filter design allowing arbitrary motion accuracy and a leaky prediction technique using both open-loop and closed-loop predictors are also introduced. The video codec described in this paper was a candidate in the competitive phase of the high-efficiency video coding (HEVC) standardization work. Compared to H.264/AVC, it demonstrated bitrate reductions of around 40% based on objective measures and around 60% based on subjective testing with 1080p sequences. It has been partially adopted into the first standardization model of the collaborative phase of the HEVC effort.**

***Index Terms*—Flexible unit representation, HEVC, video coding.**

I. Introduction

RECENTLY, with the ever increasing popularity of high-definition (HD) video content, video compression technologies which can provide higher coding efficiency than existing video coding standards have received increased attention. Early video coding standards such as MPEG-1 [1] and MPEG-2 [2] employ fixed-size block motion compensation method with a macroblock size of 16 × 16. In the state-of-the-art standard H.264/AVC [3], this macroblock size is retained but with the addition of a depth-2 quadtree, which significantly improves coding efficiency. However, it has been observed that the maximum 16 × 16 block size is too small when applied to high resolution video and causes inefficiency [4], [5]. To solve this problem, recent Visual Coding Experts Group (VCEG) proposals have extended the maximum motion block size up to 64×64 by adding additional large motion block modes on top of the conventional 16 × 16 H.264/AVC macroblock syntax [6]–[8].

Although those approaches suggested that the coding efficiency for the high resolution video materials can be improved by providing a better tradeoff between prediction performance and side information, they simply tried to add the large size motion partition while maintaining the existing 16 × 16 macroblock sizes, which imposes restrictions on the flexibility of the quadtree structure. For example, spatial/temporal prediction mode switching, intracoding tools and spatial transforms are still defined based on the 16 × 16 macroblock sizes, thus the schemes generally suffer from lower coding efficiency in intracoded blocks.

There has been extensive research on introducing a general multidepth quadtree for variable block-size motion compensation [9]–[12]. A frame is typically divided into fixed-size superblocks, with each superblock able to be further split. This kind of generic quadtree can provide a consistent codec design; however, focusing only on motion compensation does not exploit the full potential of the technique.

This paper describes a video compression framework where the whole codec has been designed to exploit flexible block sizes. A highly flexible hierarchical structure based on the generic quadtree scheme is defined by three independent block concepts: coding unit (CU), prediction unit (PU), and transform unit (TU). CU is the basic unit of region splitting. CU is analogous to the concept of macroblock, but it does not restrict the maximum size and it allows recursive splitting into four equal size CUs to improve the contents adaptivity. PU is the

Manuscript received July 23, 2010; revised October 11, 2010; accepted October 19, 2010. Date of publication November 15, 2010; date of current version January 22, 2011. This paper was recommended by Associate Editor T. Wiegand.

W. J. Han, J. Min, I. K. Kim, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. M. Hong, M. S. Cheon, N. Shlyakhov, and J. H. Park are with the Digital Media and Communication Research and Development Center, Samsung Electronics, Suwon 443-742, Korea (e-mail: wjhan.han@samsung.com; jh643.min@samsung.com; ilkoo.kim@samsung.com; elena_a.alshina@samsung.com; alexander_b.alshin@samsung.com; tammy.lee@samsung.com; jianle.chen@samsung.com; v.seregin@samsung.com; sunil.lee@samsung.com; yoonmi.hong@samsung.com; minsu.cheon@samsung.com; n.shlyakhov@samsung.com; jeonghoon@samsung.com).

K. McCann is with ZetaCast, Bath, BA2 6NZ, U.K. (e-mail: ken@zetacast.com).

T. Davies is with British Broadcasting Corporation, London, W1A 1AA, U.K. (e-mail: thomas.davies@bbc.co.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2010.2092612

1051-8215/$26.00 © 2010 IEEE

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000206

basic unit of inter/intraprediction and it may contain multiple arbitrary shape partitions in a single PU to effectively code irregular image patterns. TU is the basic unit of transform. It can be defined independently from the PU; however, its size is limited to the CU which the TU belongs to. This separation of the block structure into three different concepts allows each to be optimized according to its role, which results in the improved coding efficiency.

From these unit definitions, all coding processes can be defined in a very consistent way; large block sizes are supported in both motion compensation and intraprediction schemes, prediction mode switching is allowed at any block sizes and spatial transforms larger than 16×16 are also supported in both inter and intracoded blocks. Due to this consistency, every syntax element can be defined in exactly the same way independent of the unit sizes, which makes the specification and parsing process much simpler than would be the case with an extension scheme on top of a conventional 16×16 macroblock syntax.

Fig. 1 shows all proposed building blocks of the decoder. All the coding tools are extended to support the proposed flexible architecture. Intra prediction is extended to support an arbitrary number of angles rather than the conventional 9-modes defined in H.264/AVC. Fast integer transforms larger than 8×8 are developed to support large TU sizes. Edge definition for the deblocking filter is modified according to the CU, PU, and TU concepts. Partition information in quadtree-based adaptive loop filter (QALF) [13] is replaced with CU splitting information.

The rest of this paper is organized as follows. Section II introduces the details of the unit definitions and Section III presents the proposed coding tools for the interframe prediction. The remaining coding tools for the intraframe prediction, spatial transform, in-loop filtering, and entropy coding are introduced in Sections IV, V, VI, and VII, respectively. Section VIII summarizes the coding efficiency of the proposed techniques based on the objective and subjective results. Finally, Section IX concludes the paper. For a more complete description for the coding tools, please refer to [14] and [15].

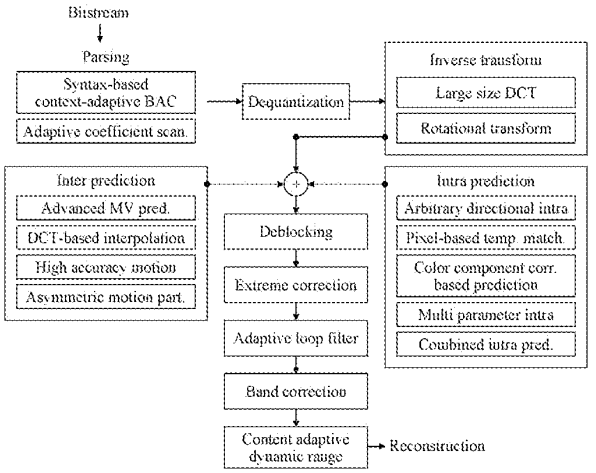


Fig. 1. All building blocks to decode the input bit-stream.

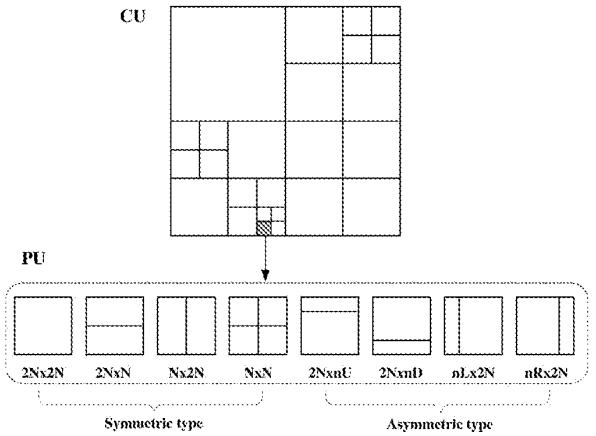


Fig. 2. Relationship between CU and PU.

II. Unit Definition

For efficient and flexible representation of video content with various resolutions, a triplet of CU, PU, and TU are introduced. The following subsections describe the details of each unit.

A. Coding Unit

The picture is divided into slices, with each slice composed of a sequence of largest coding units (LCUs). The LCU size is specified in the sequence parameter set (SPS) or slice header; the optimum LCU size is dependent on the application.

Fig. 2 shows quad-tree based CU structure. The LCU can be divided into four CUs by a 1-bit syntax, split_flag. Each split CU can be divided into four CUs recursively up to the maximum allowed hierarchical depth, also specified in the SPS or slice header. For example, if LCU size is equal to 128 and the maximum hierarchical depth is equal to 5, then 5 kinds of CU sizes are possible: 128×128 (LCU), 64×64, 32×32, 16×16, and 8×8. The three syntax elements, LCU size, maximum hierarchical depth, and split_flags, completely specify the generic multidepth quadtree.

B. Prediction Unit

Coupled with CU, this paper introduces a basic unit for prediction: the PU. The PU is defined only for the leaf node CU which is not split in quadtree structure, and the size of the PU is limited to that of the CU. The prediction method is specified by the prediction mode and the PU splitting. As with H.264/AVC, the prediction mode can be one of the values among skip, intra, and inter. Fig. 2 shows different PU splitting strategy for a CU of size $2N \times 2N$, which is dependent on the property of each prediction mode.

Skip mode can be thought of as a special intermode in which no motion vector and residue information are transmitted. Typically it is used in an area of the image with smooth motion, thus no further splitting is necessary and only $2N \times 2N$ partition is used. For the normal intermode, a set of eight splitting options are defined to cover possible motion boundaries

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000207

inside the PU. Given a PU of size $2N \times 2N$, four symmetric splittings ($2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$) and four asymmetric splittings ($2N \times$nU, $2N \times$nD, nL$\times 2N$, nR$\times 2N$) are defined. The asymmetric PU splittings are included in the design to improve the coding efficiency for the irregular object boundaries, which otherwise would be constrained to being represented less efficiently by further CU splittings to the deeper depth. For the intramode, only two possible splittings $2N \times 2N$ and $N \times N$ are defined.

All information related to the prediction is signaled on a PU basis, for example, the prediction mode and PU splitting type are specified for each PU. Once the prediction mode and PU splitting are given, motion vector difference, intraprediction direction, reference indices for each PU partition are specified on a PU partition basis.

C. Transform Unit

The TU, for transform and quantization, is the third basis unit. Although, in principle, TU can be defined in a similar way to PU to support arbitrary shape transforms, only two TU partitions are used in this paper considering the implementation complexity.

When the tu_size_flag is equal to zero, the TU size is set equal to that of the CU which it belongs to. When tu_size_flag is equal to one, the TU size is set as $N \times N$ for symmetric PU splittings and $N/2 \times N/2$ for asymmetric PU splittings, respectively. This ensures that the transform which is not applied across motion boundaries can be tested in the rate-distortion optimization process for asymmetric PU partitions.

It should be noted that the transform may be applied to the residue generated by multiple PU partitions with different motion vectors or prediction methods. This approach is especially beneficial for coding semirandom residuals generated by complex texture areas where neighboring partitions share common signal characteristics, which can exploit the better energy compaction of larger transform [14].

D. Size-Independent Syntax Representation

The proposed unit definitions allow us to design a size-independent syntax representation which specifies all syntax elements in a consistent way independent of the unit size. By contrast, in H.264/AVC and its extension in [6]–[8], block level syntax elements such as prediction_mode, transform_flag, coded_block_flag, and intra_pred_mode are coded differently depending on block size. This size-independent method of representation allows syntax elements to be specified generically for all the remaining coding tools. This property can greatly simplify the specification effort as well as the actual parsing process, especially if a large number of hierarchical depths is allowed.

III. Interprediction

In this paper, three aspects of interframe prediction are improved: motion accuracy, interpolation filter, and motion vector prediction. For motion accuracy, quarter-pel precision with 1/12-pel optional refinement is used to improve the coding efficiency while minimizing the additional overhead for signaling the motion information. For motion compensation, noncascaded spatial filtering using a discrete cosine transform (DCT)-based interpolation filter is used for every motion accuracy position. For motion vector prediction, advanced motion vector prediction (AMVP), which is a motion vector competition method [16] adapted to the flexible unit representation, is used to reduce overhead for motion information.

A. High Accuracy Motion

Even though higher motion accuracy results in less prediction error, it does not guarantee improved coding efficiency, since it leads to an increase in the bit rate to convey the motion information. For example, 1/8-pel motion accuracy may often result in a reduction in the overall coding efficiency, except for particular sequences [17].

In this paper, high accuracy motion (HAM), which adaptively refines the motion information to an arbitrary accuracy, is proposed. In the HAM method, motion information at quarter-pel accuracy is followed by a refinement flag and optional refinement information at 1/12-pel accuracy. The refinement flag is set equal to 1 only when the refinement provides the gain in the rate-distortion optimization sense, otherwise, the flag is set to 0 and no refinement information is signaled. Although the refinement information can represent any accuracy, the current design uses 1/12-pel, since refinement in 1/8-pel accuracy results in the redundancy in the refinement information.

B. DCT-Based Interpolation Filter

Motion compensation with fractional pel accuracy requires interpolation of the reference signal. In H.264/AVC standard, the prediction values at half-pel positions are obtained using a 6-tap Wiener filter, while those at quarter-pel positions are obtained by bilinear combination of the samples at integer and half-pel positions [3]. This paper proposes a DCT-based interpolation filter (DCT-IF), which can directly provide the interpolated value at the desired fractional accuracy from the samples at integer-pel positions. Since the DCT-IF can provide the interpolated value without cascaded filtering, the motion compensation process can be simplified and the complexity can also be reduced for samples at quarter-pel accuracy. Furthermore, the DCT-IF provides a unified way to generate the interpolation filters supporting motion accuracy higher than quarter-pel, e.g. 1/8-pel, 1/12-pel, etc. Note that 12-tap and 6-tap DCT-IFs are used for luma and chroma, respectively.

1) *Filter Coefficient Generation:* In this paper, trigonometric interpolation, which has been widely used for the other applications [18], is exploited to construct DCT-IF for high accuracy motion compensation. Let $p(\alpha)$ denote the sample value at the fractional position $\alpha \in [-(M-1), M]$, where M is a positive integer. Then, $p(\alpha)$ can be derived as follows:

$$p(\alpha) = \mathbf{f}_{(\alpha)}^T \mathbf{p} = \left(\mathbf{w}_{(\alpha)}^T D\right) \mathbf{p} \tag{1}$$

where $\mathbf{p} = [p_{-(M-1)}, \ldots, p_M]^T$ is the vector of the given samples at the integer-pel positions, D is the $2M \times 2M$ DCT

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000208

TABLE I
FILTER COEFFICIENTS OF 6-TAP DCT-IF

α	$2M$ = 6 (6-tap filter)						
−1/12	{	−4,	19,	254,	−19,	8,	−2 }
1/12	{	4,	−16,	252,	22,	−8,	2 }
2/12	{	6,	−28,	242,	48,	−17,	5 }
3/12	{	9,	−37,	227,	75,	−25,	7 }
4/12	{	11,	−42,	208,	103,	−33,	9 }
5/12	{	12,	−44,	184,	132,	−39,	11 }
6/12	{	11,	−43,	160,	160,	−43,	11 }
7/12	{	11,	−39,	132,	184,	−44,	12 }
8/12	{	9,	−33,	103,	208,	−42,	11 }
9/12	{	7,	−25,	75,	227,	−37,	9 }
10/12	{	5,	−17,	48,	242,	−28,	6 }

matrix whose element is given as

$$d_{ij} = \frac{1}{M}\cos\left(\frac{(2j-1+2M)i\pi}{4M}\right) \tag{2}$$

where $0 \le i \le (2M-1)$, $(-M+1) \le j \le M$, and $\mathbf{w}_{(\alpha)} = [w_0^{(\alpha)}, \ldots, w_{(2M-1)}^{(\alpha)}]^T$ is given as

$$w_i^{(\alpha)} = \begin{cases} \frac{1}{2}, & i = 0 \\ \cos\left(\frac{\pi(2\alpha-1+2M)i}{4M}\right), & 1 \le i \le (2M-1). \end{cases} \tag{3}$$

By using $\mathbf{f}_{(\alpha)}$ of (1), one can easily calculate the filter coefficients for an arbitrary number of filter taps ($2M$) and arbitrary fractional position α. Table I shows the coefficients of the 6-tap filters generated for 1/12-pel motion accuracy. Note that the filter coefficients are scaled by 256, H.264/AVC uses 5 bits for filter coefficients.

2) *Low Complexity Optimization:* An efficient hardware implementation requires multiplication-free representation for all the calculations. In order to minimize the number of arithmetical operations (shifts and additions), the filter coefficients in Table I can be slightly changed without significant loss in coding efficiency. Table II shows 5 bits representation for optimized filter coefficients for the quarter-pel positions, together with the required number of arithmetical operations. The proposed DCT-IF is a noncascaded interpolation scheme, i.e. only one spatial filtering is performed for all the possible positions. It can be implemented by a multiplication-free design. Thus, the 6-tap DCT-IF has the smaller number of operations compared to the interpolation filter of H.264/AVC since samples at the quarter-pel position can be generated by one 6-tap filtering operation whereas H.264/AVC uses a combination of 6-tap and bi-linear filters. The 12-tap DCT-IF filter requires twice as many operations as the 6-tap DCT-IF, however it provides similar performance to the well-known adaptive interpolation filter (AIF) scheme [19] while being less complex and more suitable for hardware implementation.

TABLE II
OPTIMIZED FILTER COEFFICIENTS FOR 6-TAP DCT-IF AT QUARTER-PEL POSITIONS

α	2M = 6 (6-tap filter)								Add	Shift
1/4	{	1,	−4,	28,	9,	−3,	1	}	8	2
2/4	{	1,	−5,	20,	20,	−5,	1	}	6	2
3/4	{	1,	−3,	9,	28,	−4,	1	}	8	2

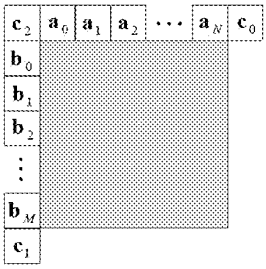


Fig. 3. Spatially adjacent motion vectors for the AMVP.

C. Advanced Motion Vector Prediction

In H.264/AVC standard, motion vectors are predicted from a median of the motion vectors of the spatially adjacent blocks [3]. To further improve the efficiency of the motion vector prediction, a few competition-based prediction methods have been proposed [16], [20], where the best predictor is selected from a given set through rate-distortion optimization.

This paper proposes the AMVP method, which is adapted to the proposed flexible unit representation. It allows the selection of the best predictor from the set which consists of three spatially adjacent motion vectors, their median, and a temporal motion vector.

Fig. 3 shows the spatially adjacent motion vectors which can be considered the candidate for the prediction. Let $A = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_N\}$, $A' \subset A$, and $I_{A'}$ denote the set of motion vectors above the current PU, the set of the available motion vectors in A, and the set which includes the indices of the elements in A', respectively. The set of the available motion vectors, A', is derived from A by verifying whether the corresponding reference index is the same as that of the current PU partition. When $A' = \emptyset$, the candidate motion vector $\mathbf{a}$ from A' is excluded from the candidate set, and a zero vector is used when taking the median of spatial motion vector predictors. Otherwise, $\mathbf{a}$ is determined as $\mathbf{a} = \mathbf{a}_n$, where n is the smallest element of $I_{A'}$. The candidate motion vectors $\mathbf{b}$ and $\mathbf{c}$ from $B = \{\mathbf{b}_0, \ldots, \mathbf{b}_M\}$ and $C = \{\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2\}$ are also determined using the same algorithm, and the resulting spatially adjacent motion vectors are used in the proposed AMVP method.

The overhead for signaling the index of the best predictor is reduced by re-organizing the set of candidate predictors. First, the predictors in the candidate set are reordered based on the PU splitting, so that the most probable motion predictor appears first in the set. Then, duplicate predictors and the predictors which can be excluded based on the parsed MVD values at the decoder side are eliminated from the set [20].

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000209

Note that, when only one predictor remains as a result of the re-organization, no overhead is signaled, since it can be also derived at the decoder.

Although AMVP is developed to predict the motion vector, it is also applied to the skip and direct modes, which do not transmit the additional motion vector differences. It should be noted that exactly the same process is applied to all cases to simplify the design. For comparison, H.264/AVC uses a slightly different scheme for the skip mode.

IV. Intraprediction

To improve the performance of intraprediction, five new intracoding tools are included. Arbitrary directional intra (ADI) extends the conventional nine-direction intraprediction to support any arbitrary number of angles in a unified way. Multiparameter intra (MPI) tries to apply post processing filters to the prediction samples from the ADI. To exploit the interchannel correlation between luminance and chrominance, color component correlation based prediction (CCCP) is introduced. Pixel-based template matching (PTM) is included to improve the coding efficiency for repeated regular patterns. Finally, the combined intraprediction (CIP) allows combining the closed-loop predictor generated from the above methods with an open-loop predictor to further improve the performance.

A. Arbitrary Directional Intra (ADI)

In H.264/AVC intracoding, nine intraprediction modes are supported for 4×4 and 8×8 blocks but only four intraprediction modes for 16×16 blocks. Intra-coded 16×16 blocks were intended to code homogeneous regions in low resolution video. However, even relatively large blocks ($\geq 16 \times 16$) cover only a small part of objects or background in high resolution video, which may contain strong directional patterns.

Arbitrary directional intra (ADI) is a coding tool designed to extend the directional intraprediction scheme used in H.264/AVC to give more accurate representation of directions and better suit larger prediction units. In the ADI scheme, up to 30 angles are generated using the integer pair (dx, dy) to represent the direction of lines which each mode uses for context pixel extrapolation as illustrated in Fig. 4.

Like H.264/AVC, ADI uses the reconstructed pixels. However, in some cases, the down-left region is available for the prediction process as depicted in Fig. 4. When the down-left region is not available, that region is generated using the nearest reconstructed pixel value.

Due to the large variety of the possible unit sizes, it is beneficial to adjust the number of angles according to the unit size. For example, PU partitions with 64×64 and 128×128 sizes are usually used for homogeneous areas which can be generated by using only some prediction modes, such as DC and plane modes. In this case, the number of supported prediction modes of the current PU partition may differ from that of the neighboring PU partitions. Intra prediction modes of the neighboring PU partitions are converted to the nine most frequently used modes to predict the current intraprediction mode in a unified way.

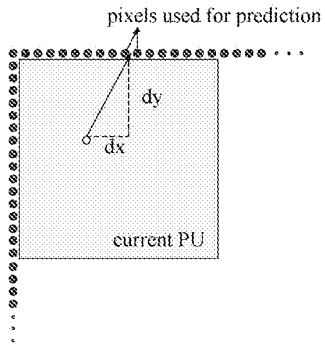


Fig. 4. Neighboring pixels and angles defined by (dx, dy).

B. Multiparameter Intra

Directional intraprediction sometimes results in nonsmooth and unnatural patterns which are not transform-friendly. This problem can be alleviated by applying a smoothing filter to the prediction signal obtained by the ADI scheme described in Section IV-A. Multiparameter intra (MPI) allows the use of the 4-point filter in (4) for each pixel to increase the probability of generating a prediction signal well-suited to the following transform stages. For each PU partition, mpi_flag is used to specify whether or not the MPI scheme is applied.

$$p'[x, y] = (p[x, y] + p[x-1, y] + p[x, y-1] + p[x, y+1] + 2) >> 2 \quad (4)$$

where $p[x, y]$ and $p'[x, y]$ are the original and the smoothed prediction value at (x, y) coordinates, respectively, and $>>$ is the bit-wise shift to the right.

C. Color Component Correlation Based Prediction

There have been some studies on predicting chrominance pixels using the correlation with luminance [21]. Although most approaches try to predict the pixel values of one color channel by another color channel, the color component correlation based prediction (CCCP) scheme derives an object segmentation map of the chrominance channel from the already coded luma channel, and applies it to intraprediction.

The CCCP procedure is as follows. Firstly, the size of the reconstructed luma samples including the neighboring luma pixels are adjusted to match that of the chroma samples according to the given chroma format. Then, simple thresholding using the mean value is exploited to generate the binary segmentation map. After that, two mean values are computed from the neighboring chroma pixels according to the segmentation map and assigned to the inner part of the chroma block. Finally, a 3×3 averaging filter is applied to improve the smoothness of the prediction. Fig. 5 shows a brief flow of the CCCP process.

D. Pixel Based Template Matching (PTM)

Whilst directional extrapolation based intraprediction tools are very efficient for regions with directional patterns, they are not well-suited to regions with repeated regular patterns. To overcome this limitation, several intratechniques using some

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000210

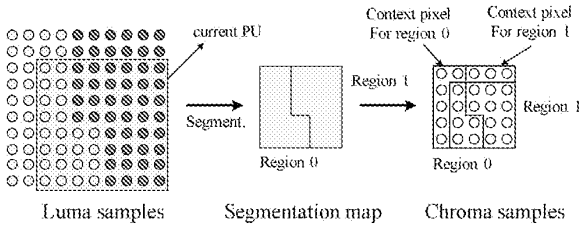


Fig. 5. Processing flow of color component correlation based prediction (CCCP).

form of template matching based on the already decoded pixels have been proposed [22], [23]. Although the reported schemes were proven to provide good coding gains for the targeted areas, a major drawback was the high complexity at the decoder side.

In this paper, a simple pixel based template matching (PTM) approach is proposed. PTM is designed to apply template matching at the pixel level to maximize the degrees of freedom. PTM uses only three pixels as template. Search candidates are also restricted to neighboring 27 pixels, which results in complexity reduction in matching process. Another difference of PTM compared with previous methods is that already prediction samples in the same prediction unit are also used as template and search candidate. This allows use very limited search area even for large block prediction.

E. Combined Intraprediction

Combined intraprediction (CIP) is a coding tool for providing improved prediction, especially in large blocks where a directional linear prediction may not work well, even with the large range of angles available to ADI. CIP predictions comprise a weighted combination of the prediction samples provided by other intracoding tools together with a pixel-by-pixel local mean prediction of pixels within the block to be transformed.

The local mean is constructed as the average of pixel values to the left, top-left and above the current pixel. In the decoder and within the local decoder in the encoder, these values are the reconstructed values after inverse quantization and transform. In the encoder, the reconstructed values are not available in the interior of the block because this would require quantization, transform and inverse transform to have been performed, and the pixels used for prediction lie within the block. Instead original values are used, although the encoder may select any suitable values. This local mean prediction is therefore different between forward prediction (encoder only) and reconstruction (encoder and decoder). Due to this property, CIP can be seen as a leaky prediction method to combine the conventional closed-loop predictor and an open-loop predictor. The advantage of CIP is that it provides pixel-by-pixel adaptation but is a simpler tool than, for example, local template matching approaches.

The combination factor is a 5-bit number between 0 and 31/32 which determines the proportion of local mean in the prediction. Since this number is less than 1, the prediction/reconstruction difference noise is damped, and pixel-by-pixel prediction adaptation is possible without noise blow-up. The ADI component provides an overall prediction direction and the local mean prediction provides local adaptation.

V. Spatial Transforms

It is well known that transforms larger than 8×8 provides better energy compaction and less quantization error for smooth and coarse textured video areas at the expense of computational complexity [24], [25]. In this paper, 16×16, 32×32 and 64×64 fast integer transforms are presented in addition to the conventional 4×4 and 8×8 transforms to support large TUs. In addition, a rotational transform (ROT) tool, which can adaptively compact the energy into low frequency components by rotating the transform basis, is also proposed as a second transform.

A. Large Integer Transform (16×16, 32×32, and 64×64)

The DCT is independent of input signal while it still provides near optimal decorrelation properties. Moreover, many fast DCT algorithms have been developed. All large transforms proposed in this paper are scaled integer DCTs based on Chen's fast factorizations [26]. Chen's fast DCT has a regular butterfly structure which is beneficial to hardware implementation.

Similarly to the H.264/AVC 4×4 core transform, each multiplication factor in the flow graph of the fast algorithm is scaled and approximated by some fixed precision values p, i.e., each constant is approximated by the dyadic rational of the form $q/2^p$, which can be implemented with only additions and bit-shifting operations. The precision value p can be chosen according to the dynamic range limitation and coding efficiency. As all multiplication factors are approximated, the core transform may not be truly orthogonal. However, this minor nonorthogonality does not result in any perceptible negative effect on the compression performance, whilst the complexity can be significantly reduced. Furthermore, it is still possible to achieve perfect reconstruction using a lifting scheme that can be applied additionally to the butterfly structure [27].

After all multiplication factors are determined, the forward 2-D transform is performed separately in vertical and horizontal direction as follows:

$$Y = C_f(C_f X^T)^T = C_f X C_f^T \tag{5}$$

where X is $N \times N$ residual data, C_f is the core N-point forward transform, and Y is $N \times N$ transformed coefficients. The calculation of $C_f X^T$ is performed from left to right in the flow graph of the fast algorithm. For the inverse transform, the computation order is reversed from right to left. Since the core transform C_f is scaled DCT, the remaining scaling factors should be integrated into the quantization stage.

B. Rotational Transform

While DCT is the most widely used block transform for video and image codecs, the DCT basis functions are not optimal for some types of residual signal. For example, a residual signal having strong diagonal components cannot

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000211

be represented efficiently with the DCT basis vectors. Typically, directional transform schemes are proposed to exploit this problem such as mode dependent directional transform (MDDT) [28], [29].

However, the direct use of directional transform is not appropriate to the proposed flexible unit definitions since it is difficult to design the directional transform by a fast butterfly structure such as DCT. In addition, the number of transforms would be almost doubled because both DCT and the directional transform for various TU sizes would be kept in the codec design.

In this paper, ROT is proposed as a second transform that could be applied after the core integer transform instead of introducing completely new transform cores. The main idea of ROT is to change the coordinate system of the transform basis, instead of direct rotation of the input residual. For example, the rotation matrices of horizontal and vertical directions for a 4×4 block with the rotation angles of $(\alpha_1, \alpha_2, \alpha_3)$ and $(\alpha_4, \alpha_5, \alpha_6)$ are defined as

$$R_{\text{ver}}(\alpha_1, \alpha_2, \alpha_3) = \begin{pmatrix} C_1C_3 - S_1C_2S_3 & -S_1C_3 - C_1C_2S_3 & S_2S_3 & 0 \\ C_1S_3 + S_1C_2C_3 & -S_1S_3 + C_1C_2C_3 & -S_2C_3 & 0 \\ S_1S_2 & C_1S_2 & C_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6)$$

$$R_{\text{hor}}(\alpha_4, \alpha_5, \alpha_6) = \begin{pmatrix} C_4C_6 - S_4C_5S_6 & -S_4C_6 - C_4C_5S_6 & S_5S_6 & 0 \\ C_4S_6 + S_4C_5C_6 & -S_4S_6 + C_4C_5C_6 & -S_5C_6 & 0 \\ S_4S_5 & C_4S_5 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

where $C_k = \cos\alpha_k$ and $S_k = \sin\alpha_k$ for $k = 1, \ldots, 6$. The rotation matrices for 8×8 block can be defined similar to those for 4×4 block.

Each set of angles $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\}$ specifies one element in the ROT dictionary. In ROT, only four out of all possible rotational angles sets are used. These RT matrices minimize the discrepancy between forward-inverse RT in integer arithmetic realization and provide better energy compaction. These sets of rotational angles were selected using the training with VCEG test sequences, and they provide similar performance for other video sequences including the test sequences for the HEVC standardization. This proves that these angles sets capture and exploit general characteristic of natural images. The index of rotational angles in the dictionary is signaled to the decoder for each CU explicitly.

Furthermore, since ROT is a second transform, not all of the DCT coefficients are processed. For TUs larger than 4×4, only 8×8 low-frequency areas are processed. It has been found that this restriction does not degrade the coding efficiency since most of the coefficients are already compacted into these low-frequency areas.

The final 2-D spatial transform process when the ROT is used can be summarized as

$$Y = R_{\text{ver}}(C_f X C_f^T) R_{\text{hor}} \quad (8)$$

where X is residual data, C_f is the core forward transform, and Y is output coefficients.

VI. In-Loop Filtering

After reconstruction, several in-loop filters are applied to reduce the distortion between the original and the reconstructed video frames. In addition to a modified H.264/AVC deblocking filter for the CU/PU/TU structure, a CU-synchronized adaptive loop filter (CU-ALF) is introduced to reduce the average distortion of the reconstructed frame. Extreme correction (EXC) and band correction (BDC) techniques are applied to reduce the distortion further by considering object-based pixel classifications. Finally, content adaptive dynamic range (CADR) is proposed to reduce the performance loss from the rounding effects of the various internal processes.

A. Deblocking Filter

The same process as used in H.264/AVC is employed for the deblocking filter. However, the use of a large size transform can effectively reduce the complexity of the deblocking filter, by reducing the number of deblocked pixels. It was reported that the number of deblocked pixels was reduced by 43% on average [14] for the 1080p sequences used in [30].

B. CU-Synchronized Adaptive Loop Filter (CU-ALF)

QALF [13] is known to provide significant coding gain by effectively combining both the Wiener filter approach and the quadtree based on/off selections. In QALF, along with filter coefficients, the information representing the quadtree structure is coded independently with the prediction quadtree structure, which results in large amount of side information in the slice header.

Since the proposed unit representations provides a generic quadtree structure similar to the original design of QALF, this information can be reutilized for the adaptive loop filter process. CU-synchronized adaptive loop filter (CU-ALF) is based on the QALF approach; however, it uses the already coded CU hierarchy as the on/off control for the adaptive loop filter process instead of coding separate quadtree information in the slice header.

It has been reported that the proposed CU-ALF provides comparable coding gain to QALF, while eliminating the need for any side information related to the quadtree partition in the slice header and reducing the encoder complexity [14].

C. Extreme Correction and Band Correction

EXC is designed to compensate the distortion according to the identified pixel classes. Each pixel is categorized as local maxima, local minima, or object edge by simple computation based on the four neighboring pixel values: left, above, right, and bottom. For each pixel class, the mean difference between the original and the reconstructed signals is computed and transmitted to the decoder. A fuller description is given in [14].

BDC is another form of pixel classification, based on pixel value ranges. The main motivation is to equalize the different probability distribution functions of the reconstructed signal with the original. In this paper, up to 16 value ranges are defined; the mean difference for each band is computed and transmitted to the decoder.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000212

D. Content Adaptive Dynamic Range

Internal bit depth increase (IBDI) is a technique to reduce the rounding error by enlarging the internal data precision [31]. Typically, IBDI is implemented by increasing the source bit-depth, which makes the encoder implementation more difficult. CADR is related to IBDI, but is a more general form of source scaling. Using the CADR, a pixel value x is adaptively scaled to $F(x)$ as follows:

$$F(x) = (2^{\text{bit}} - 1)(x - \text{Min})/(\text{Max} - \text{Min}) \tag{9}$$

where bit, Max, and Min are the bit-depth, the maximum value, and the minimum value of the input source, respectively. It should be noted that the parameters Min and Max can be adjusted at sequence, slice, or even block level.

CADR can be used to reduce the rounding error without requiring increased bit-depth in the encoder. For example, source material that is known to be constrained by the [16, 235] safe range of BT.709 can be mapped to [0, 255] to reduce rounding errors in the internal representation.

VII. Entropy Coding

In this paper, syntax-based context-adaptive binary arithmetic coder (SBAC) is used for the entropy coding of syntax elements. Additionally, adaptive coefficient scanning (ACS), which provides multiple scanning patterns for the transform coefficients, is also proposed to improve the coding efficiency while reducing the complexity.

A. Syntax-Based Context-Adaptive Binary Arithmetic Coding

As with the context-adaptive binary arithmetic coding (CABAC) of H.264/AVC [32], the SBAC consists of four major steps: syntax-based binarization, context modeling, probability estimation, and binary arithmetic coding. The syntax-based binarization and the context modeling are adapted to the syntax elements obtained from the flexible unit representation and the extended coding tools. For example, the assignment of context models for the syntax elements related to transform coefficients (e.g. significance map, coefficient level, etc.) is dependent on the hierarchical depth of the CU. The probability estimation and binary arithmetic coding are based on an improved version of the arithmetic coding method specified in Annex D of the JPEG still image coding standard [33]. Specifically, the table-based probability update and the multiplication-free design are improved, and the bypass mode for equi-probable binary symbols is added to reduce complexity [14].

B. Adaptive Coefficient Scanning

Since large transforms frequently have very sparse coefficient distributions, a fixed zigzag scanning method may result in long coefficient scanning lists including many zeroes, resulting in inefficient coding. ACS that adaptively chooses the best scanning method for each TU is used to avoid this problem.

Given the quantized transform coefficients, the best scanning pattern is chosen from conventional zigzag, horizontal, and vertical scanning patterns, and then its index is explicitly signaled in each TU. Note that the ACS index is signaled only when nonzero AC coefficients exist in the TU, since there is no difference between the patterns when the TU includes only the DC coefficient.

When the horizontal or vertical scanning methods are chosen as the best scanning method, the mapping process from the coefficient array indices to the scanning order indices is not necessary. Thus, the complexity for the coefficient indexing can be reduced and also the memory bandwidth to access the zigzag mapping table can be saved.

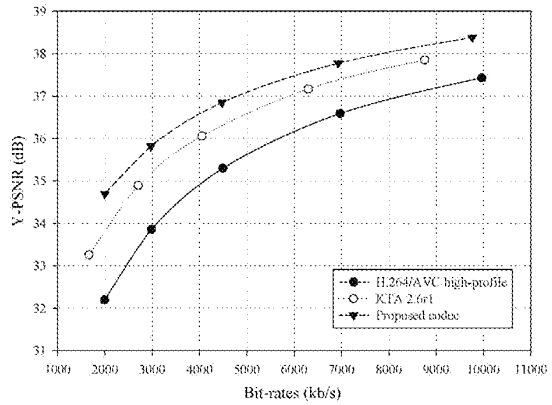


Fig. 6. Rate distortion curve of BasketballDrive 1080p sequence.

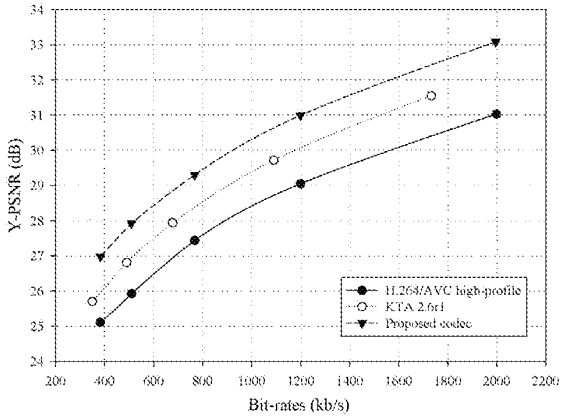


Fig. 7. Rate distortion curve of PartyScene WVGA sequence.

VIII. Performance Analysis

To verify the performance of the proposed coding tools, they were implemented into one complete video codec software which is available at [14]. This section provides both objective performance based on the peak signal-to-noise ratio (PSNR) measure and the subjective performance based on the mean opinion score (MOS) values for the proposed video codec.

A. Objective Compression Efficiency

1) *Overall Performance:* The coding efficiency of the proposed system is verified using two sets of coding configurations defined in the Call for Proposals [30]—constraint

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000213

set 1 (CS1) and constraint set 2 (CS2) which correspond to a random access case and a low delay case, respectively. Figs. 6 and 7 show typical examples of rate-distortion curves when video sequences are coded using CS1 configuration. As shown in the figures, the proposed codec consistently showed significant coding benefits over H.264/AVC High Profile. For comparison with other recent advances in the video coding area, the results of the key technology area (KTA) 2.6r1 [34] with the same test configurations are also included. This test configuration is same as in [35], where all major tools of KTA including enhanced AIF, MV competition, MDDT, high precision filter, ALF, and extended block size are enabled. It should be noted that the improved coding efficiency can be observed with both high resolution and low resolution video material.

Table III summarizes the performance of the proposed codec relative to the H.264/AVC High Profile. The average compression performance improvement over all sequences is 39.49% in CS1 and 39.48% in CS2, respectively. Class B (1080p resolution) achieves 43.29% (CS1) and 43.66% (CS2) bit saving on average, with BQTerrace as the best sequence in both configurations, showing 54.06% (CS1) and 60.62% (CS2) performance improvement, respectively. RaceHorses in Class D (WQVGA resolution) shows the lowest performance improvement in both configurations, at 27.08% in CS1 and 21.59% in CS2, respectively. The performance summarized in Table III shows that the proposed codec consistently achieves the excellent coding performance regardless of the resolution of input video sequences and the coding configuration.

2) *Performance of Each Set of Coding Tools:* It is hard to evaluate the performance of each individual coding tool, since the coding tools interact with each others and the tool-wise evaluation cannot measure the performance improvement introduced by the interaction. Therefore, the proposed coding tools are categorized to five sets—large transform (16×16, 32×32, and 64×64 transforms), AMVP, inter (DCT-IF and HAM), intra (ROT, MPI, ADI, CCCP, TMI, and CIP), and in-loop (EXC, ALF, and CADR)—and the performance of each set of coding tools is evaluated by measuring the performance loss introduced by turning off the tools in the corresponding set. The performance of the large motion partition is not evaluated since its performance has been already verified in the literatures [6]–[8]. Table IV summarizes the performance of each set of coding tools and the alternative setting which is enabled when the corresponding tool set is turned off. Note that only 1-second tests using CS1 configuration were carried out to reduce the experiment time. The results show that all the sets of coding tools contribute to the overall performance improvement almost equally.

TABLE III
SUMMARIZED BD-RATE REDUCTIONS FOR VARIOUS RESOLUTIONS

Class (Size)	Sequence Name	CS1 (%)		CS2 (%)	
		BD-Rate	Average	BD-Rate	Average
A (4K)	*Traffic*	40.59	35.21	N/A	N/A
	PeopleOnStreet	29.84		N/A	
B (1080p)	*Kimono*	44.42	43.29	44.45	43.66
	ParkScene	32.96		31.18	
	Cactus	40.78		34.74	
	BasketballDrive	44.21		47.29	
	BQTerrace	54.06		60.62	
C (WVGA)	*BasketballDrill*	43.09	39.94	36.82	35.58
	BQMall	39.69		37.06	
	PartyScene	41.13		40.45	
	RaceHorses	35.86		27.99	
D (WQVGA)	*BasketballPass*	30.55	36.45	26.93	32.16
	BQSquare	54.19		51.87	
	BlowingBubbles	33.97		28.24	
	RaceHorses	27.08		21.59	
E (720p)	*Vidyo 1*	N/A	N/A	48.38	47.48
	Vidyo 3	N/A		46.26	
	Vidyo 4	N/A		47.79	
Total average		39.49		39.48	

TABLE IV
PERFORMANCE OF EACH SET OF CODING TOOLS

Tool Set	Alternative Setting	Class (%)				
		A	B	C	D	Average
Large tr.	Max tr. size = 8×8	2.2	7.2	1.9	1.0	3.5
AMVP	Median predictor [3]	4.4	3.7	2.9	2.4	3.2
Inter	6-tap/bi-linear filter [3]	0.4	1.3	4.2	7.6	3.6
Intra	Directional intra [3]	3.5	3.3	4.4	2.4	3.4
In-loop	Deblocking only	8.8	6.1	3.6	2.9	4.3

B. Subjective Compression Efficiency

A large scale subjective test was carried out for all the proposals submitted to the Call for Proposals. In this test, it was shown that the proposed codec provided excellent subjective improvements over H.264/AVC in all configurations and it gave one of the best overall results of all of the proposals that were submitted, based on MOS values. The official report of subjective test results can be found in [36].

Table V shows several pairs of MOS values for H.264/AVC High Profile and the proposed codec for the five 1080p sequences. Each pair has been selected to give similar MOS values, to verify the bit-rate reduction that the proposed codec provides for similar subjective visual quality. As shown in the table, the proposed codec operating at an average of 2.4 Mbit/s provides similar visual quality to H.264/AVC High Profile operating at an average of 6.4 Mbit/s. This corresponds to approximately 60% bit-rate savings, indicating that the proposed codec provides even greater improvement in subjective visual quality than would be expected from objective metrics such as PSNR.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000214

TABLE V
SUMMARIZED MOS VALUES FOR 1080P SEQUENCES

Sequence	H.264/AVC		Proposed		
	Bit rates (kb/s)	MOS	Bit rates (kb/s)	MOS	Reduction (%)
Kimono	4000	8.86	1600	8.56	60.0
ParkScene	4000	7.41	1600	7.78	60.0
Cactus	7000	8.61	3000	8.24	57.1
BasketballDrive	7000	7.44	3000	8.17	57.1
BQTerrace	10000	8.77	3000	9.17	70.0
Average	6400	8.22	2440	8.38	60.9

IX. CONCLUSION

In this paper, a new flexible unit representation composed of CU, PU, and TU was proposed. The proposed scheme provides a generic quadtree structure while defining the relationship between region splitting, prediction method, and transform sizes with minimal signaling overhead.

In addition, various other coding tools were extended in a very consistent way according to the proposed unit representation. For example, the number of directions in the directional intraprediction scheme was extended to arbitrary angles and the supported transform size was extended up to 64×64.

By combining other novel coding tools designed under the proposed unit representation, the proposed codec showed about 40% performance improvement over the state-of-the-art H.264/AVC standard based on objective PSNR measurements. The large-scale subjective testing results show that an average of about 60% coding gain can be achieved for the 1080p sequences.

The video codec described in this paper was a candidate in the competitive phase of the HEVC standardization work and it has been partially adopted into the first standardization model of the collaborative phase of the HEVC effort.

ACKNOWLEDGMENT

The authors would like to thank their colleagues including, but not limited to, H. K. Jung, J. C. Lee, S. R. Lee, D. Y. Kim, K. H. Lee, Y. M. Sohn, Y. Y. Lee, B. K. Lee, S. M. Park, H. S. Song, J. B. Choi, B. D. Choi, C. H. Lee, S. S. Jung, and D. S. Cho for their invaluable contributions and continuous comments on this paper.

REFERENCES

[1] *Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s: Video*, ISO/IEC 11172-2, ISO/IEC JTC1/SC29/WG11, Aug. 1993.

[2] *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13818-2, ISO/IEC JTC1/SC29/WG11, 1996.

[3] *Final Draft International Standard of Joint Video Specification*, ITU-T Rec. H.264, ISO/IEC 14496-10, ISO/IEC JTC1/SC29/WG11, Mar. 2003.

[4] K. H. Lee, E. Alshina, J. H. Park, W. J. Han, and J. H. Min, "Technical considerations for ad hoc group on new challenges in video coding standardization," in *Proc. 85th MPEG Meeting*, Jul. 2008, no. M15580.

[5] E. Alshina, K. H. Lee, W. J. Han, and J. H. Park, "Technical considerations for ad hoc group on new challenges in video coding standardization," in *Proc. 86th MPEG Meeting*, Oct. 2008, no. M15899.

[6] P. Chen, Y. Ye, and M. Karczewicz, "Video coding using extended block sizes," in *Proc. 36th VCEG Meeting*, Oct. 2008, no. VCEG-AJ23.

[7] T. Yoshino, S. Naito, and S. Sakazawa, "Preliminary response for draft call for evidence on high performance video coding," in *Proc. 87th MPEG Meeting*, Feb. 2009, no. M16082.

[8] S. Sekiguchi and S. Yamagishi, "On coding efficiency with extended block size for UHDTV," in *Proc. 87th MCEG Meeting*, Feb. 2009, no. M16019.

[9] A. Puri, H. M. Hang, and D. Schilling, "An efficient block-matching algorithm for motion-compensated coding," in *Proc. ICASSP*, vol. 12. Aug. 1997, pp. 1063–1066.

[10] M. H. Chan, Y. B. Yu, and A. G. Constantinides, "Variable size block matching motion compensation with applications to video coding," *Commun. Speech Vis. IEE Proc. I*, vol. 137, no. 4, pp. 205–212, Aug. 1990.

[11] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," *IEEE Trans. Image Process.*, vol. 3, no. 3, pp. 327–331, May 1994.

[12] J. J. Zhang, M. O. Ahmad, and M. N. Swamy, "Quadtree structured region-wise motion compensation for video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 5, pp. 808–822, Aug. 1999.

[13] T. Chujoh, N. Wada, and G. Yasuda, *Quadtree-Based Adaptive Loop Filter*, document VCEG-C181, ITU-T Study Group 16, Jan. 2009.

[14] K. McCann, W. J. Han, I. K. Kim, J. H. Min, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. M. Hong, M. S. Cheon, and N. Shlyakhov, "Samsung's response to the call for proposals on video compression technology," in *Proc. 1st JCT-VC Meeting*, Apr. 2010, no. JCTVC-A124.

[15] JCT of ISO/IEC MPEG & ITU-T VCEG, "Test model under consideration," in *Proc. 1st JCT-VC Meeting*, Apr. 2010, no. JCTVC-A205.

[16] J. Jung and G. Laroche, "Competition-based scheme for motion vector selection and coding," in *Proc. 29th VCEG Meeting*, Jul. 2006, no. VCEG-AC06.

[17] T. Wedi, "Hybrid video coding based on high-resolution displacement vectors," in *Proc. VCIP*, Jan. 2001, pp. 186–193.

[18] C. C. Tseng and S. L. Lee, "Design of fractional delay FIR filter using discrete cosine transform," in *Proc. APCCAS*, Nov. 2008, pp. 858–861.

[19] T. Wedi and H. G. Musmann, "Motion- and aliasing-compensated prediction for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 577–586, Jul. 2003.

[20] S.-D. Kim and J. B. Ra, "An efficient motion vector coding scheme based on minimum bitrate prediction," *IEEE Trans. Image Process.*, vol. 8, no. 8, pp. 1117–1120, Aug. 1999.

[21] S. H. Lee and N. I. Cho, "Intra prediction method based on the linear relationship between the channels for YUV 4:2:0 intra coding," in *Proc. ICIP*, 2009, pp. 1037–1040.

[22] T. K. Tan, C. S. Boon, and Y. Sujuki, "Intra prediction by template matching," in *Proc. ICIP*, 2006, pp. 1693–1696.

[23] J. Balle and M. Wien, "Extended texture prediction for H.264/AVC intra coding," in *Proc. ICIP*, 2007, pp. VI-93–VI-96.

[24] S. Ma and C.-C. J. Kuo, "High-definition video coding with supermacroblocks," in *Proc. VCIP*, Jan. 2007.

[25] J. Dong, K. N. Ngan, C. K. Fong, and W. K. Cham, "2-D order-16 integer transforms for HD video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 10, pp. 1462–1474, Oct. 2009.

[26] W.-H. Chen, C. H. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. 25, no. 9, pp. 1004–1009, Sep. 1977.

[27] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3032–3044, Dec. 2001.

[28] Y. Ye and M. Karczewicz, *Improved Intra Coding*, Contribution C257, ITU-T Q.6/SG16, Geneva, Switzerland, Jun. 2007.

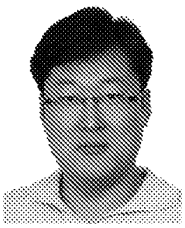
[29] Y. Ye and M. Karczewicz, *Improved Intra Coding*, VCEG Contribution VCEG-AG11, no. 20, Shenzhen, China, Oct. 2007.

[30] ISO/IEC JTC1/SC29/WG11, "Joint call for proposals on video compression technology," in *Proc. 91st MPEG Meeting*, Apr. 2010, no. N11113.

[31] T. Chujoh and R. Noda, "Internal bit depth increase for coding efficiency," in *Proc. 31st VCEG Meeting*, Jan. 2007, no. VCEG-AE13.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000215

[32] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.
[33] *Coding of Still Pictures*, ISO/IEC IS 10918-1, ITU-T Rec. T.81, ISO/IEC JTC1/SC29/WG1, Sep. 1993.
[34] ITU-T VCEG. (2010). *Key Technology Area (KTA) Software* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/KTA
[35] T. K. Tan, G. Sullivan, and T. Wedi, *Recommended Simulation Common Condition for Coding Efficiency Experiments Revision 1*, VCEG Contribution VCEG-AJ10, Jul. 2008.
[36] V. Baroncini, J. R. Ohm, and G. Sullivan, "Report of subjective test results of responses to the joint call for proposals (cfp) on video coding technology for high efficiency video coding (HEVC)," in *Proc. 1st JCT-VC Meeting*, Apr. 2010, no. JCTVC-A204.



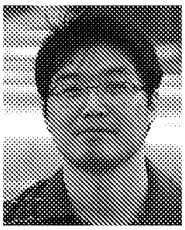
Woo-Jin Han (M'02) received the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1997 and 2002, respectively.

He is currently a Principal Engineer with the Multimedia Platform Laboratory, Digital Media and Communication Research and Development Center, Samsung Electronics, Suwon, Korea. Since 2002, he has contributed successfully to the ISO/IEC Moving Pictures Experts Group, Joint Video Team, and Joint Collaborative Team standardization efforts. His current research interests include high-efficiency video compression techniques, scalable video coding, multi-view synthesis, and visual contents understanding.



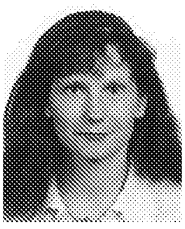
Junghye Min received the B.S. degree in mathematics from Ewha Women's University, Seoul, Korea, in 1995, and the M.E. and Ph.D. degrees in computer science and engineering from Pennsylvania State University, University Park, in 2003 and 2005, respectively.

From 1995 to 1999, she was a S/W Engineer with Telecommunication Systems Business, Samsung Electronics, Suwon, Korea. She is currently a Principal Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics. Her current research interests include video content analysis, motion tracking, pattern recognition, and video coding.



Il-Koo Kim received the M.S. and Ph.D. degrees from the School of Electrical Engineering, Seoul National University, Seoul, Korea, in 2001 and 2006, respectively.

After joining the Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea, in 2006, he has worked to develop high efficiency video codec. His current research interests include video compression and transmission.



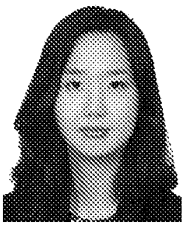
Elena Alshina received the M.S. degree in physics from Lomonosov Moscow State University (MSU), Moscow, Russia, in 1995. She continued her postgraduate study at the Faculty of Computational Mathematics and Cybernetics, MSU, and received the Ph.D. degree in mathematical modeling in 1998.

Since that time, she has been with the Institute for Mathematical Modeling, Russian Academy of Sciences, Moscow, performing research on high accuracy numerical methods and new finite-difference scheme development. Simultaneously, she was an Associate Professor for the Moscow Institute of Electronic Technology, Moscow. In 2006, she joined Samsung Electronics (first at the Moscow Research Center, before moving to Korea). Currently, she is a Senior Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea, where she makes high efficiency video codecs.



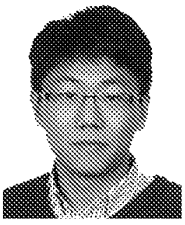
Alexander Alshin received the M.S. and Ph.D. degrees in mathematical physics from Moscow State University (MSU), Moscow, Russia, in 1995 and 1998, respectively.

From February 1998 to January 2006, he was a Senior Researcher with the Physical Department, MSU. Since 2006, he has been with Samsung Electronics, Suwon, Korea, and currently works with the Multimedia Platform Laboratory, DMC Research and Development Center, Samsung Electronics, as a Senior Engineer. His current research interests include theory of non-classical PDEs, nonlinear functional analysis, applied mathematics, numerical analysis, and their applications to video compression.



Tammy Lee received the B.S. degree in mathematics education from Ewha Women's University, Seoul, Korea, in 2000, and the M.S. degree in computer science from New York University, New York, in 2005.

She is currently a Senior Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea. Her current research interests include image and video compression.



Jianle Chen received the B.S. and Ph.D. degrees in information and communication system from Zhejiang University, Hangzhou, China, in 2001 and 2006, respectively.

Currently, he is a Senior Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea. His current research interests include video/image compression, computer vision, and image/video processing.



Vadim Seregin received the M.S. (honors) degree in physics from Moscow State University (MSU), Moscow, Russia, and the Ph.D. degrees in mathematical modeling, and numerical methods and software systems from MSU, in 2002 and 2005 respectively.

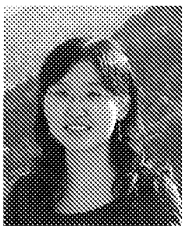
Currently, he is a Senior Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea. His current research interests include quasi-gasdynamics, numerical methods, and the field of image and video processing with a focus on video compression.



Sunil Lee (S'02–M'08) received the B.S. degree in electrical engineering from Yonsei University, Seoul, Korea, in 2001, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2002 and 2008, respectively.

From March 2008 to August 2008, he was a Post-Doctoral Researcher with the Division of Electrical Engineering, School of Electrical Engineering and Computer Science, KAIST. He is currently a Senior Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea. His current research interests include multimedia retrieval, multimedia security, digital watermarking, multirate signal processing, and video coding.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000216



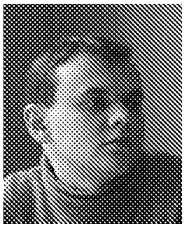
Yoon Mi Hong received the B.S. degree in mathematics from Yonsei University, Seoul, Korea, in 2003, and the M.S. degree in applied mathematics and the Ph.D. degree in mathematical sciences from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2005 and 2009, respectively.

She is currently a Senior Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea. Her current research interests include time-frequency analysis, sampling theory, and its applications in signal processing, as well as video coding.



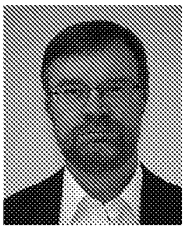
Min-Su Cheon received the B.S. and M.S. degrees in electrical engineering from Kyunghee University, Suwon, Korea, in 2005 and 2007, respectively.

He is currently an Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon. His current research interests include high-efficient video coding, multiview video coding, 3-D video coding, scalable video coding, and bioimage processing.



Nikolay Shlyakhov received the M.S. degree in applied mathematics from the Moscow Institute of Physics and Technology (MIPT), Moscow, Russia, in 1999. From 1999 to 2007, he was pursuing the Ph.D. degree from the Institute for Mathematical Modeling of the Russian Academy of Sciences, Moscow.

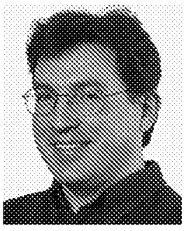
From 2003 to 2007, he was an Assistant Professor with the Department of Higher Mathematics, MIPT. Since 2007, he has been an Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea. His current research interests include numerical mathematics, theory of approximation, and video coding.



Ken McCann graduated from Glasgow University, Glasgow, U.K., with the B.S. degree in natural philosophy in 1979. He received the M.S. degree in systems engineering from Surrey University, Surrey, U.K., in 1982.

He is currently a Director and Co-Founder of ZetaCast, Bath, U.K., an independent U.K.-based technology consultancy company specializing in digital TV and related systems. Prior to founding ZetaCast, he was the Technology Director with NTL Broadcast (now Arqiva), the main provider of broadcasting infrastructure in the U.K. Before that, he was with Symbionics Instruments, Cambridge, U.K., the Independent Broadcasting Authority and Philips Research Laboratories. He contributed significantly to the development of the MPEG-2 standard, has chaired the DVB technical group responsible for audio-visual coding specification since its inception over 15 years ago, and is currently an active member of the Joint Collaborative Team on Video Coding, working on the new high efficiency video coding standard.

Mr. McCann has been a member of the IET in the U.K. (and its predecessor, the IEE) since 1982.



Thomas Davies graduated from Oxford University, Oxford, U.K., with the M.A. degree in mathematics and philosophy in 1991. He received the M.S. and Ph.D. degrees in mathematics from Warwick University, Coventry, U.K., in 1992 and 1996, respectively.

He is currently a Lead Research and Development Engineer with Research and Development, British Broadcasting Corporation, London, U.K., having joined in 2000. He specializes in video compression, and image and signal processing, and was responsible for developing the open source Dirac video compression algorithms, the intra coding part of which is standardized as SMPTE VC-2. He is actively participating in the development of the MPEG high efficiency video coding standard under the auspices of the Joint Collaborative Team on Video Coding. Prior to joining the BBC, he was a Technology Consultant in systems engineering and satellite networking.



Jeong-Hoon Park received the B.S. and M.S degrees in electrical engineering from Hanyang University, Seoul, Korea, in 1991 and 1994, respectively.

He is currently a Principal Engineer with the Multimedia Platform Laboratory, Digital Media and Communication Research and Development Center, Samsung Electronics, Suwon, Korea. In 1998, he was a Visiting Scholar with the University of California, Los Angeles, where he researched error resilient video compression and transmission. Since then, he has contributed to the ITU-T Video Coding Experts Group and ISO/IEC Moving Pictures Experts Group. From 2002 to 2004, he has been involved in developing mobile broadcasting technology as an active member of WorldDMB forum. His current research interests include audio and video compression technology, medical image processing, and multimedia systems including video broadcasting and communications.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000217

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13666680	BICI ET AL.
	Examiner	Art Unit
	CLIFFORD HILAIRE	2488

✓	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	06/26/2015								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								
	19	✓								
	20	✓								

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000218

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-10)
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 contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	13666680
Filing Date	2012-11-01
First Named Inventor	Mehmet Oguz BICI
Art Unit	2631
Examiner Name	TBD
Attorney Docket Number	NC77198US-NP

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20120307905	A1	2012-12-06	Sunyeon et al.	
	2	20110170602	A1	2011-07-14	Tammy et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1	2011062392	WO	A2	2011-05-26	SK Telecom Co. Ltd. et al.		☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /CH/

Receipt date: 04/22/2013 13666680 - GAU: 2488

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**	Application Number	13666680
	Filing Date	2012-11-01
	First Named Inventor	Mehmet Oguz BICI
	Art Unit	2631
	Examiner Name	TBD
	Attorney Docket Number	NC77198US-NP

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	International Search Report and Written Opinion received for corresponding Patent Cooperation Treaty Application No. PCT/FI2012/051070, dated March 27, 2013, 12 pages.	☐
	2	Oudin, S. et al.; "Block Merging for Quadtree-Based Video Coding", IEEE Int. Conf. on Multimedia and Expo, July 11-15, 2011, 6 pages.	☐
	3	Sullivan, G. J.; "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 22, No. 12, December 2012, pp. 1649-1668.	☐

If you wish to add additional non-patent literature document citation information please click the Add button **Add**

EXAMINER SIGNATURE

Examiner Signature	/CLIFFORD HILAIRE/	Date Considered	06/24/2015

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /CH/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**	Application Number	13666680
	Filing Date	2012-11-01
	First Named Inventor	Mehmet Oguz BICI
	Art Unit	2631
	Examiner Name	TBD
	Attorney Docket Number	NC77198US-NP

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☐ Fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ None

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Thomas J. Arria/	Date (YYYY-MM-DD)	2013-04-22
Name/Print	Thomas J. Arria	Registration Number	60223

This collection of information is required by 37 CFR 1.97 and 1.98. 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 1 hour 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.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /CH/

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000221

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 the Freedom of Information Act requires disclosure of these record s.

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 inspections 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.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /CH/

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S3	9	((Mehmet) near2 (BICI)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/24 17:00
S4	85	((Jani) near2 (LAINEMA)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/24 17:00
S5	59	((Kemal) near2 (UGUR)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/24 17:00
S6	4	("20110170602" | "20120307905").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/06/25 09:41
S7	2	("20110170602" | "20120307905").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 09:41
S8	9	((Mehmet) near2 (BICI)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 09:43
S9	85	((Jani) near2 (LAINEMA)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 09:43
S10	59	((Kemal) near2 (UGUR)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 09:43
S11	104	S8 S9 S10	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 09:43
S12	45	S11 and merg$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 09:43
S13	12	S11 and merg$4 near list$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 09:44
S14	45	S11 and merg$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 09:44
S15	18	S11 and merg$4 same candidate	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:10
S16	67	S11 and candidate	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:11
S17	63	S11 and candidate with motion	US-PGPUB; USPAT;	OR	ON	2015/06/25 10:13

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000223

			USOCR			
S18	55	S11 and candidate with motion with vector	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:13
S19	45	S11 and candidate with motion with vector same block	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:14
S20	28	S11 and (candidate and list and vector)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:15
S21	5805	S11 and (candidate and list and vector)".clm"	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:15
S22	4	S11 and (candidate and list and vector).clm.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:15
S23	1	"20130114723".pn.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:52
S24	1	"20130114723".pn. and (candidate with vector with motion) same (merg$4 set list subset)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:53
S25	1812	(candidate with vector with motion) same (merg$4 set list subset)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:55
S26	27634	((H04N19/52 OR H04N19/513 OR H04N19/176 OR H04N19/61 OR H04N19/51 OR H04N19/597 OR H04N19/56 OR H04N19/139 OR H04N19/573 OR H04N19/521 OR H04N19/103 OR H04N19/30).CPC.)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 10:57
S27	1	"20130034162".pn.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:08
S28	1	"20130034162".pn. and (candidate with motion with vector)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:16
S29	1	"20130034162".pn. and (candidate with motion with vector) same block	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:17
S30	1	"20130034162".pn. and (candidate)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:18
S31	1	"20130034162".pn. and (candidate) and (list set subset)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:19
S32	1	"20130034162".pn. and (candidate same block) and (list set subset)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:19
S33	1274	(candidate with vector with motion) same (merg$4 set list subset) and (motion vector) with search$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:32
S34	1064	(candidate with vector with motion) same (merg$4 set list subset) and (motion vector) near4 search$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:32
S35	888	(candidate with vector with motion) with (merg$4 set list subset) and (motion	US-PGPUB; USPAT;	OR	ON	2015/06/25 11:32

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000224

		vector) near4 search$4	USOCR			
S36	618	(candidate with vector with motion) near4 (merg$4 set list subset) and (motion vector) near4 search$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:33
S37	112	(candidate with vector with motion) near4 (merg$4 set list subset) same (motion vector) near4 search$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:33
S38	92	(candidate with vector with motion) near4 (merg$4 set list subset) same (motion vector) near4 search$4 and "375".clas.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:33
S39	150592	(candidate with vector with motion) near4 (merg$4 set list subset) same (motion vector) near4 search$4 same block "375".clas.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:33
S40	51	(candidate with vector with motion) near4 (merg$4 set list subset) same (motion vector) near4 search$4 same block and "375".clas.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:34
S41	8	(candidate with vector with motion) near4 (merg$4 set list subset) same (motion vector) near4 search$4 same block and subset and "375".clas.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:42
S42	2	(candidate with vector with motion) near4 (merg$4 set list subset) same (motion vector) near4 search$4 same block and subset and (remov$4 exclud$4) with (list set merg$4 subset)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 11:45
S43	50	("5719627" "5815602" "4989089" "5428403" "5781249" "5838391" "4851906" "4989087" "5235419" "5371549" "5414469" "5457481" "5506622" "5521642" "5532746" "5546129" "5557341" "5574663" "5579050" "5581308" "5587741" "5598216" "5612743" "5614954" "5617144" "5619281" "5625417" "5627591" "5638129" "5642166" "5646867" "5654761" "5689306" "5694487" "5717463" "5719630" "5724369" "5731851" "5751362" "5862261" "5903672" "5991447" "6005980" "6020925" "6023298" "6026195" "6057884" "6091777" "6154519" "6205177").pn.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 13:29
S44	25	S43 and (candidate same motion)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 13:30
S45	0	S43 and (candidate same motion) and merg$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 13:30
S46	4	S43 and (candidate same motion) and exclud$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 13:31
S47	4	S43 and (candidate) with (vector motion) and exclud$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/25 13:32
S48	3	(US-20130034162-$ or US-20120106645-$ or US-20120106638-	US-PGPUB	OR	ON	2015/06/25 13:44

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000225

		$).did.				
S49	0	S48 and "PU "	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/06/25 13:44
S50	2	S48 and prediction with unit	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/06/25 13:44
S52	9	((Mehmet) near2 (BICI)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:02
S53	85	((Jani) near2 (LAINEMA)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:02
S54	59	((Kemal) near2 (UGUR)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:02
S55	104	S52 S53 S54	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:02
S56	55	S55 and candidate with motion with vector	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:02
S57	10	S55 and candidate with motion with vector and transcod$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:03
S58	8	S55 and candidate with motion with vector and transcod$4 and encod$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:04
S59	53	S55 and candidate with motion with vector and encod$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:05
S60	29	S55 and candidate with motion with vector same encod$4	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:05
S61	24	S55 and candidate with motion with vector same encod$4 and unit and (block macroblock)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:07
S62	18	S55 and candidate with motion with vector same encod$4 and (unit near predict$5) and (block macroblock)	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:10
S63	23	S55 and prediction near unit	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:11
S64	22	S55 and prediction near unit and candidate	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:11
S65	872	prediction near unit and candidate near4 motion near4 vector	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:14
S66	713	prediction near unit and candidate near4	US-PGPUB;	OR	ON	2015/06/26

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000226

		motion near4 vector same block	USPAT; USOCR			06:14
S67	653	prediction near unit and candidate near4 motion near4 vector with block	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:14
S68	172	prediction near unit same candidate near4 motion near4 vector with block	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:14
S69	42	prediction near unit same candidate near4 motion near4 vector with block same encoded	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:15
S70	32	("prediction unit" PU) with block with encoded same candidate near4 motion near4 vector	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:17
S71	483	PU and candidate with motion with vector	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:35
S72	209	PU same candidate with motion with vector	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:36
S73	160	PU same candidate with motion with vector same block	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:36
S74	142	PU same candidate with motion with vector same block and "375".clas.	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 06:36
S75	1	(US-20120134415-$).did.	US-PGPUB	OR	ON	2015/06/26 07:05
S76	0	S75 and encoded	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 07:05
S77	0	S75 and encod$4 with block	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 07:05
S78	1	S75 and coded with block	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 07:05
S79	126	PU same candidate with motion with vector same block and merge with list	US-PGPUB; USPAT; USOCR	OR	ON	2015/06/26 07:06
S80	1	(US-20120008688-$).did.	US-PGPUB	OR	ON	2015/06/26 08:47
S81	1	(US-20120008688-$).did.	US-PGPUB	OR	ON	2015/06/26 08:52
S82	1	(US-20120008688-$).did. and (PU prediction near unit)	US-PGPUB	OR	ON	2015/06/26 08:53
S83	1	(US-20120008688-$).did. and (PU prediction near unit) and receiv$4	US-PGPUB	OR	ON	2015/06/26 08:54
S84	1	(US-20120008688-$).did. and (PU prediction near unit) and (encod$4 cod$4 decod$4 receiv$4)	US-PGPUB	OR	ON	2015/06/26 08:55

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000227

S51	0	"20130034162".pn. and (candidate)	USPAT; UPAD	OR	ON	2015/06/25 11:18

6/ 26/ 2015 1:09:19 PM
C:\ Users\ chilaire\ Documents\ EAST\ Workspaces\ 13666680.wsp

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000228

A Multi-Pass True Motion Estimation Scheme With Motion Vector Propagation for Frame Rate Up-Conversion Applications

Shen-Chuan Tai, Ying-Ru Chen, Zheng-Bin Huang, and Chuen-Ching Wang

***Abstract*—This paper presents a robust true motion estimation algorithm, designated as MPMVP (Multi-pass and Motion Vector Propagation), to enhance the accuracy of the motion vector fields in frame rate up-conversion applications. The MPMVP uses a multi-pass scheme to progressively refine approximate motion vectors to true motion vectors based upon the motion information acquired in previous passes. The multi-pass motion estimation process uses a large block size to detect the motion vectors within the objects themselves and small block sizes to detect the motion vectors along the object boundaries. Actually, the block size is progressively reduced during the search process. When the motion vector of a block is considered to be sufficiently accurate for motion estimation purposes, the block is said to be converged and the local motion vector search process terminates. A novel technique, referred to as motion vector propagation, is then applied to propagate the motion vector of the converged block to its neighboring blocks. This technique not only ensures that neighboring motion vectors within the same object have a high degree of spatial correlation, but also accelerates the convergence of the motion vectors in the neighboring blocks and therefore reduces the overall computational time and expense of the multi-pass motion vector search procedure. A novel distortion criterion is proposed to enhance the tolerance of the traditional sum-of-absolute-difference measurement technique applied in the motion estimation scheme to noise and shadow effects. The experimental results demonstrate that the proposed true motion estimation algorithm outperforms the traditional full search, 3DRS and TCSBP algorithms in terms of both the smoothness of the generated motion vector fields and the visual quality of the up-converted frames.**

***Index Terms*—Candidate motion vector, distortion criterion, frame rate up-conversion (FRUC), recursive search, true motion, variable block-size.**

I. Introduction

MODERN video formats have a wide variety of resolutions, aspect ratios, frame rates, and so forth. Typically, the frame rates of capturing devices vary between 24 and 60 Hz. However, the frame rates used by most display devices tend to range from 50 to 120 Hz, for example. Therefore, when playing back video sequences, it is necessary to interpolate some additional frames based on the original video sequence using some form of frame rate up-conversion (FRUC) algorithm. However, traditional schemes such as picture repetition, linear interpolation and other non-motion compensated methods generally produce visual artifacts such as trembling or blurring in the interpolated frames. The quality of the interpolated frames can be improved by combining the FRUC scheme with motion estimation (ME) techniques. Many complex algorithms for interpolating frames based upon the motion information of objects between successive frames have been proposed [1]–[15]. However, they tend to be highly time consuming and computationally expensive.

The ME schemes used in FRUC algorithms differ from the conventional ME techniques applied in coding standards. Generally speaking, ME algorithms used in frame rate up-conversion applications are termed "true ME" schemes in order to emphasize the importance of obtaining accurate motion vectors for all of the objects in the frame. True ME is employed not only in FRUC schemes, but also in such diverse applications as video encoding, [16], surveillance system tracking [17], de-interlacing schemes [18], error concealment in decoding systems [19], [20], and so forth. Briefly, the objective of video coding systems is to minimize the motion compensated residual. In block-matching ME algorithms, the features of moving objects are tracked using distortion criteria such as the sum of absolute difference (SAD) or the sum of square error (SSE). However, video sequences tend to contain high levels of noise, and, therefore, the motion vectors (MVs) extracted by minimal SAD/SSE-based coding systems do not always accurately reflect the true motions of the objects in the frame. In FRUC schemes, it is essential to obtain accurate motion vectors to ensure the quality of the interpolated frames, and therefore the ME algorithms used in such schemes require a robust ability to deal with the effects of noise. Unfortunately, the ME algorithms conventionally used in block-matching coding systems, e.g., full search, logarithmic search, diamond search, and so on, fail to address this requirement.

In frame rate up-conversion, the optimal image quality is obtained when the objects in the frames are interpolated along their respective motion trajectories. 3-D recursive search block matching (3DRS) [4] and temporal compensated motion estimation with simple block-based prediction (TCSBP) [2] have proposed ME algorithms based on predictive motion vectors (PMVs) to accelerate the block-matching speed and to enhance the degree of spatial correlation between neighboring blocks. In [3], the authors used an overlapped block-based motion estimation (OBME) scheme possessing enhanced ME capabil-

Manuscript received August 8, 2006; revised November 12, 2006, April 23, 2007, and October 20, 2007.

S.-C. Tai, Y.-R. Chen, and Z.-B. Huang are with the Department of Electrical Engineering, National Cheng Kung University, Tainan 701, Taiwan, R.O.C. (e-mail: cyr91d@dcmc.ee.ncku.edu.tw).

C.-C. Wang is with the Institute of Electrical Engineering, National Changhua University of Education, Changhua City 500, Taiwan, R.O.C.

Color versions of one or more figures are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JDT.2007.916014

1551-319X/$25.00 © 2008 IEEE

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000229

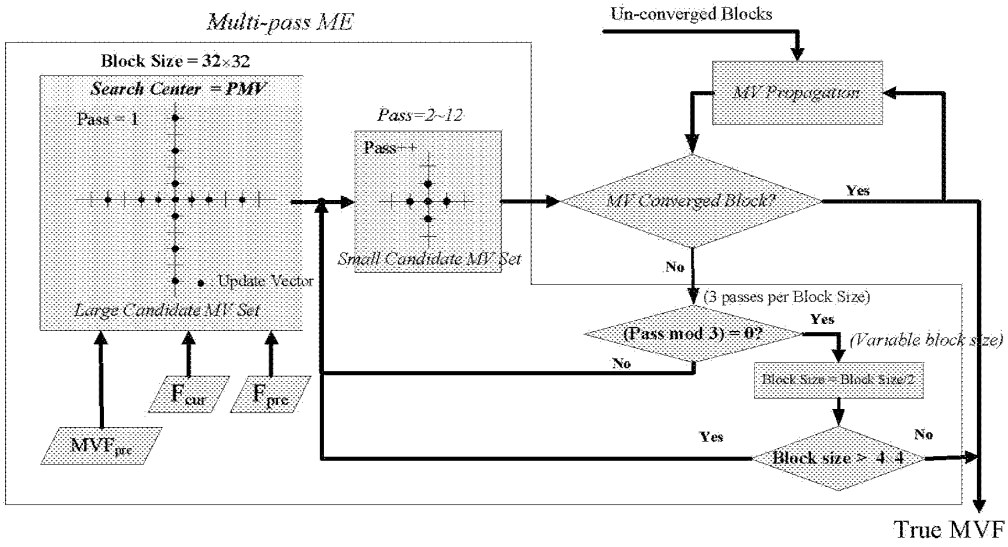


Fig. 1. Flow diagram of overall system.

ities and the ability to detect global motion through the use of a large-block matching mechanism. However, without accurate and sufficient information regarding the object shape, OBME schemes are unable to predict the motion of the object boundaries with any degree of accuracy, and therefore result in an incorrect motion aggregation in the motion vector field (MVF). A fundamental requirement of all true ME schemes is to achieve a high degree of spatial motion correlation inside a frame since all of the blocks within the same object must necessarily move in the same direction. Although several algorithms have been presented [4]–[6], [13] which employ some form of penalty function to enhance motion correlation, they are incapable of resolving the problem of incorrect motion propagation. In [10], [11], the MVs were established using hierarchical block-matching algorithms and were then propagated between hierarchical levels based on the regions of the objects. However, obtaining the regions of the objects requires a time-consuming segmentation process and the accuracy of the segmentation results cannot be guaranteed. In [14], a variable block size technique was used to enhance the ME accuracy. In this approach, the MVs were initially found for small blocks and were then used as the criteria for merging neighboring blocks. Having merged neighboring blocks, the MV search procedure was then repeated. However, this method is not only computationally expensive, but also achieves only limited success since small blocks cannot provide sufficient information regarding the global characteristics of an object and thus the reliability of the merging process cannot be ensured.

Accordingly, this paper presents a robust true ME scheme for FRUC applications. The remainder of this paper is organized as follows. Section II presents an overview of the major components of the proposed scheme, including the multiple-pass, the MV convergence decision technique, the MV convergence propagation scheme, and the distortion criterion used to suppress the effects of noise and shadow in the interpolated frames. Section III presents the experimental results obtained by the proposed true ME scheme for the MVFs of several well-known video test sequences, and compares the performance of the proposed scheme with that of the full search, 3DRS and TCSBP schemes when applied to frame rate up-conversion applications. Finally, Section IV presents some brief conclusions.

II. Multi-Pass Motion Estimation Strategy and MV Propagation Rule for True Motion Estimation

The proposed ME scheme searches for the true MVs using a multi-pass strategy in which the block size is progressively reduced from 32×32 to 4×4. After each pass, a convergence decision rule is applied to determine whether or not the MV identified for the block is sufficiently accurate. If the MV has acceptable accuracy, the block is said to be converged and the local MV search procedure is terminated. As indicated above, the largest block size used in the multi-pass ME scheme is 32×32. When an object is bigger than this block size, a technique referred to as MV propagation can be applied to spread the MV of the converged block to all of the blocks within the object. Fig. 1 presents an overall view of the multi-pass MV search procedure and associated MV propagation routine.

A. Multi-Pass Motion Estimation Strategy

The basic principle of the multi-pass ME strategy proposed in this study is to utilize the motion information relating to neighboring blocks generated in the previous pass to gradually refine the accuracy of the MV in the current block. The multi-pass search scheme executes a maximum of 12 passes. In the first three passes, the MV search is performed using a block size of 32×32. However, in the next three passes, the block size is reduced to 16×16, while in the final two sets of three passes, the block size is further reduced to 8×8 and 4 × 4, respectively. The MV for block $\vec{X}$ found in frame t in the ith pass is defined as $\overrightarrow{\mathrm{MV}}^{i}(\vec{X}, t)$. The MV search in block $\vec{X}$ in the i th pass is performed in accordance with (1). As shown, the search is confined to the vectors $\vec{V}$ defined in the candidate vector set $CS^{i}(\vec{X}, t)$, i.e., a full search is not performed. As discussed later, the term

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000230

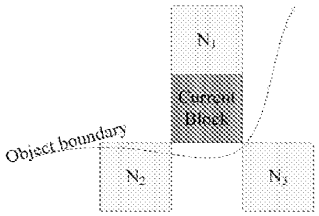


Fig. 2. Geometric relation between motion edge and block position.

$e(\vec{V}, \vec{X}, t)$ in (1) is a new distortion criterion based upon the conventional SAD distortion criterion.

$$\overrightarrow{MV}^{i}(\vec{X}, t) = \arg \min_{\vec{V} \in CS^{i}(\vec{X}, t)} (e(\vec{V}, \vec{X}, t)), \qquad 1 \le i \le 12. \tag{1}$$

The candidate vector set $CS^i(\vec{X}, t)$ is comprised by a single PMV and a set of update vectors. Let $\vec{X}$ represent the block location and let t be the frame number. The candidate vector set for this particular block in the ith pass are then defined as

$$CS^{i}(\vec{X}, t) = \left\{ \vec{V} \,|\, \vec{V} = \overrightarrow{\text{PMV}}^{i}(\vec{X}) + \vec{\text{U}}, \vec{\text{U}} \in US^{i} \right\} \tag{2}$$

where $\overrightarrow{\text{PMV}}^{i}(\vec{X})$ is the PMV of block $\vec{X}$ and is chosen from the MVs of the neighboring blocks and US^i is a set of update vectors. Every motion vector $\vec{V}$ in the candidate vector set is comprised by a single PMV and a update vector $\vec{U}$ selected from update vector set US^i.

1) PMV: Although the MVs of all eight neighboring blocks can theoretically be used as PMV predictors for the current block, evaluating each of these predictors in order to establish the most suitable PMV among them is computationally expensive and time consuming. Therefore, this study uses the geometric relationship proposed in [2] describing the correlation between the current block position and the motion edge to reduce the number of PMV predictors to be evaluated. As shown in Fig. 2, at least one of the three neighboring blocks and the current block must belong to the same object no matter where the object boundary crosses, and, therefore, the number of PMV predictors which must be considered can be reduced from 8 to just 3. The scheme presented in [2] involved a single ME pass, and therefore employed a single PMV predictor pattern. However, as described above, the current search scheme executes a maximum of 12 passes. In general, different PMV predictor patterns may result in different levels of accuracy. Therefore, the current multi-pass ME scheme considers 12 different PMV predictor patterns, as shown in Fig. 3.

¶ 4

Each pass in the proposed ME scheme is performed using one of the PMV predictor patterns (see Fig. 3) and then one MV in the selected PMV pattern which has minimal distortion is selected as the final PMV of this pass. In traditional single-pass ME algorithms, the current block has no information regarding the motion of the succeeding blocks because their MVs are not obtained yet. However, in multi-pass ME algorithms, the motion information of all blocks in current pass can be referred in future passes.

¶ 5

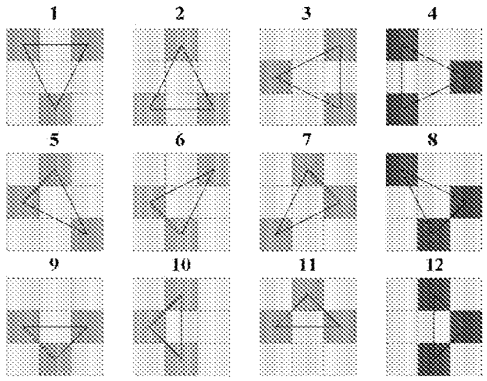


Fig. 3. PMV predictor patterns.

2) Update Vector Set: In (2), the term US^i represents the set of update vectors in the ith pass and has the form

$$US^{i} = \begin{cases} US_{\text{large}}, & i = 1 \\ US_{\text{small}}, & 2 \le i \le 12 \end{cases}$$
$$US_{\text{small}} = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \pm 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \pm 1 \end{pmatrix} \right\}$$
$$US_{\text{large}} = US_{\text{small}} \cup \left\{ \begin{pmatrix} \pm 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \pm 3 \end{pmatrix}, \begin{pmatrix} \pm 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \pm 5 \end{pmatrix} \right\}. \tag{3}$$

The update vector sets employed in the proposed algorithm vary between different passes. The proposed scheme uses a large update vector set (i.e., a greater number of larger MVs) in the initial passes when employing a large block size to detect coarse motions in the frame. When the update vectors have a large range, e.g., ±(5,5), finding good approximations to the true MVs in blocks which cross object boundaries is straightforward. Furthermore, in the first pass, the motion information of neighboring blocks is simply copied from the previous frame and therefore there is no guarantee that this information is reliable. As a result, the PMV obtained for the block is not necessarily an accurate representation of the true MV. However, using a large update vector set increases the probability of obtaining a good approximation to the true MV. In later passes of the search procedure, a highly precise PMV with a small update vector set (i.e., fewer and smaller vectors) is used to refine these initial non-true motion estimates into true motion estimates.

As indicated earlier, the term $e(\vec{V}, \vec{X}, t)$ in (1) is a new distortion criterion based on the conventional SAD distortion criterion and has the form

$$e(\vec{V}, \vec{X}, t) = \sum_{i,j \in \text{block}} \text{weightAD}(\vec{V}, \overrightarrow{X_{ij}}, t)/\text{blocksize}$$
$$\text{AD}(\vec{V}, \overrightarrow{X_{ij}}, t) = \text{abs}(I(\overrightarrow{X_{ij}}, t) - I(\overrightarrow{X_{ij}} + \vec{V}, t - T))$$
$$\text{weight} = \begin{cases} 0, & \text{AD}(\vec{V}, \overrightarrow{X_{\text{ij}}}, t) < 5 \\ 1, & 5 \le \text{AD}(\vec{V}, \overrightarrow{X_{\text{ij}}}, t) < 30 \\ 2, & \text{otherwise} \end{cases} \tag{4}$$

where $I(\vec{X}_{\text{ij}}, t)$ is the gray level of a pixel located at coordinates (i,j) in block $\vec{X}$ of frame t and abs indicates absolute value. According to Weber's law, human eyes have a noise threshold

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000231

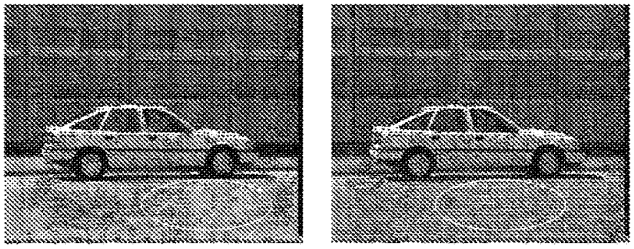


Fig. 4. MVFs of frames with Gaussian noise: (a) MVF produced by proposed algorithm with traditional SAD distortion criterion, and (b) MVF produced by proposed algorithm with new distortion criterion.

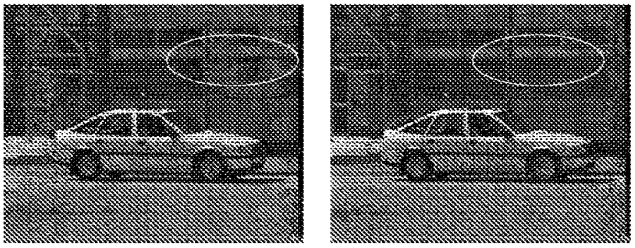


Fig. 5. Up-converted frames from VECTRA sequence. (a) MVF produced by 3DRS with penalty function. (b) MVF produced by 3DRS with proposed distortion criterion.

of approximately 5 grayscale value (for contrast) in pictures. For experience, an edge often leads to a gradient greater than 30 grayscale value. Therefore, if the absolute difference (AD) between current pixel and referred pixel in previous frame is less than 5, the human eye will ignore the distortion. However, if it is greater than 30, the distortion will be very noticeable. Hence, as shown in (4), the distortion criterion applied in the current ME search scheme has the form of a piecewise linear function. In general, noise has a particularly serious effect on the accuracy of the MV search process in smooth areas of the frame, i.e., background area. Therefore, the piecewise linear function in (4) allows small distortions (AD $<$ 5) caused by noise to be ignored while simultaneously placing more emphasis on the large distortions (AD $\geq$ 30) caused by the edges, and therefore the accuracy of the MV is enhanced.

Fig. 4 shows the MVFs obtained by the proposed ME scheme using the conventional SAD distortion criterion and the new distortion criterion given in (4), respectively, for a frame with Gaussian noise. Comparing the two frames, it is apparent that the improved noise suppression characteristics of the proposed distortion criterion yield more accurate MVs vectors in the smooth regions of the image, i.e., on the grass. Fig. 5 presents two up-converted frames from the VECTRA test sequence obtained by the 3DRS search scheme using the original penalty function and the new distortion criterion, respectively. It can be seen that the MVF produced by 3DRS using the new distortion criterion is more smooth. For example, in Fig. 5(b) more motion vectors of the window area indicate to the same direction than in Fig. 5(a). In other words, the proposed distortion criterion is applicable not only to the multi-pass scheme presented in the current study, but also to existing true ME algorithms.

B. Decision Rules for MV Converged Block

To reduce the computational complexity and expense of the proposed ME search scheme, processing within a local block terminates as soon as the block is converged, irrespective of the number of passes which have been completed. Furthermore, the degree of motion correlation among neighboring blocks is enhanced by using a MV propagation technique that will be described in Section II-C. The degree of spatial motion correlation between neighboring blocks is determined in accordance with

$$\Delta\mathrm{MV}(\overrightarrow{C},\overrightarrow{N}) = \mathrm{abs}(\mathrm{MV}(\overrightarrow{C},t).x - \mathrm{MV}(\overrightarrow{N},t).x) + \mathrm{abs}(\mathrm{MV}(\overrightarrow{C},t).y - \mathrm{MV}(\overrightarrow{N},t).y) \quad (5)$$

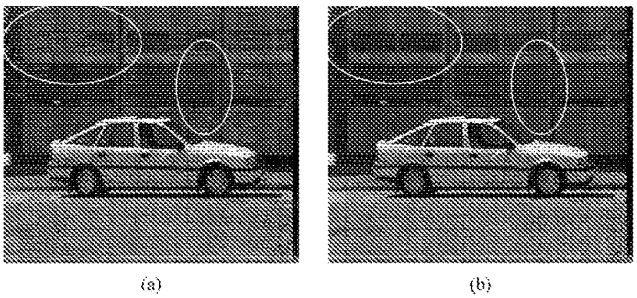


Fig. 6. (a) Proposed ME scheme without convergence propagation and (b) proposed ME scheme with convergence propagation.

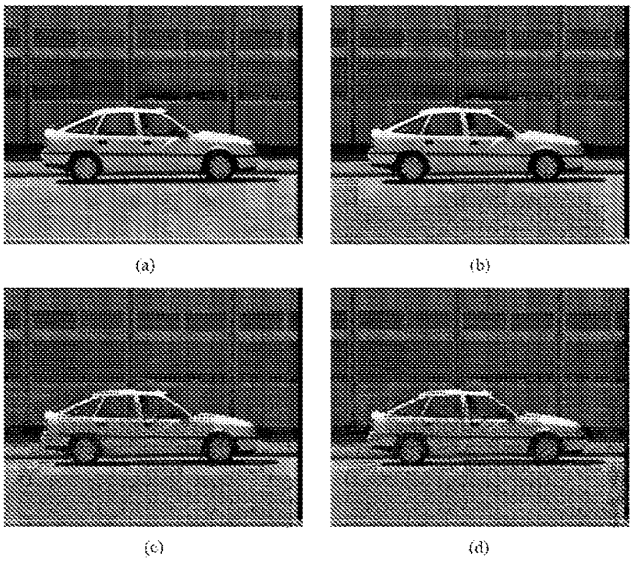


Fig. 7. Motion vectors of converged Blocks (MV is shown by block-size 4×4). (a) 3rd Pass: block size is 32 × 32, (b) 6th Pass: block size is 16×16. (c) 9th Pass: block size is 8×8. (d) 12th Pass: block size is 4×4.

where $\overrightarrow{C}$ and $\overrightarrow{N}$ are the current block and a neighboring block in spatial motion correlation test range, respectively, and x and y are the horizontal and vertical MV components. True motion is obtained when the distortion value is low, i.e., $e(\overrightarrow{C},\overrightarrow{X},t) < T_1$ and the degree of motion correlation between the current block and all of its adjacent blocks in the spatial motion correlation

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000232

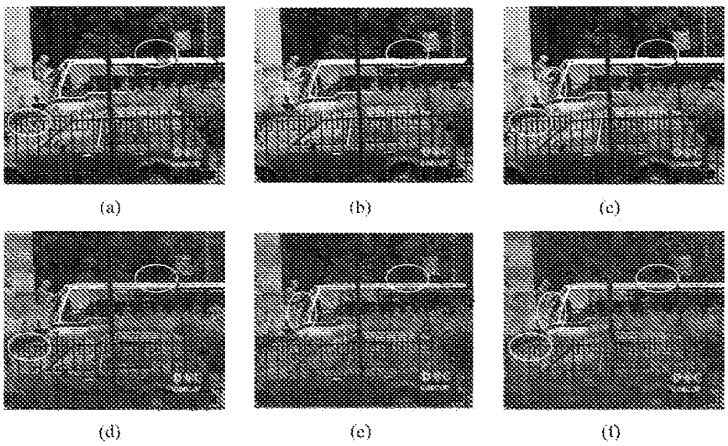


Fig. 8. (a), (d) Interpolated frame from BUS sequence and corresponding MVF obtained from 3DRS, (b), (e) Interpolated frame from BUS sequence and corresponding MVF obtained from TCSBP. (c), (f) Interpolated frame from BUS sequence and corresponding MVF obtained from proposed MPMVP algorithm.

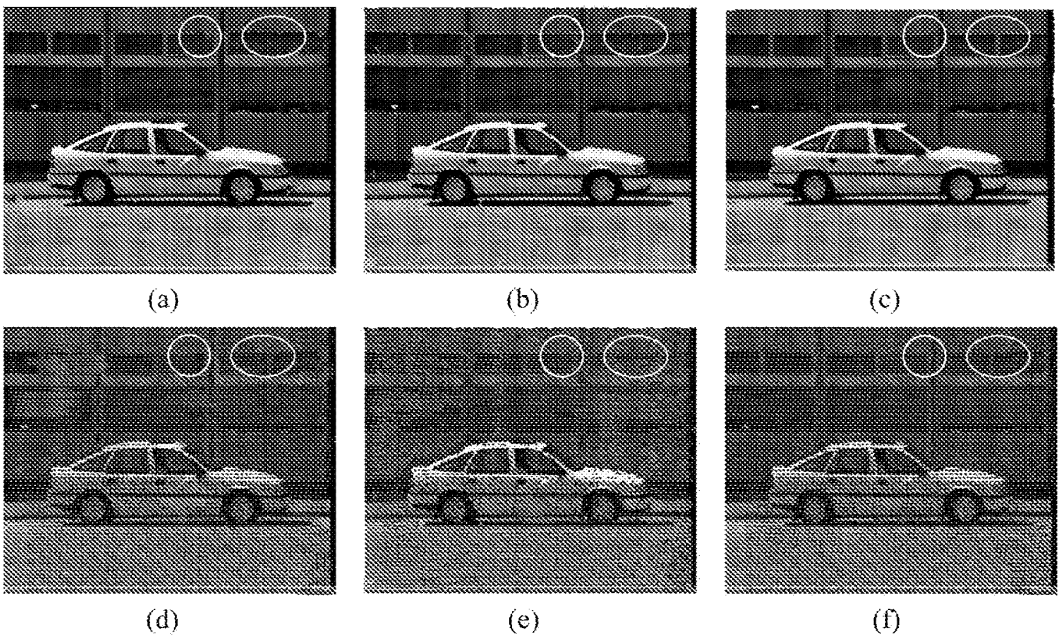


Fig. 9. (a), (d) Interpolated frame from VECTRA sequence and corresponding MVF obtained from 3DRS. (b), (e) Interpolated frame from VECTRA sequence and corresponding MVF obtained from TCSBP. (c), (f) Interpolated frame from VECTRA sequence and corresponding MVF obtained from proposed MPMVP algorithm.

test range is high, i.e., $\Delta\mathrm{MV}(\overrightarrow{C}, \overrightarrow{N}) < T_2$, where T_1 is a distortion threshold and T_2 is a threshold value designed to enable sufficient activity for the MV search to prevent the clustering of inappropriate MVs. In other words, the current scheme considers both the distortion value and the degree of motion correlation when searching for true MV. This approach differs from existing ME algorithms, which typically terminate the MV evaluation process as soon as an acceptable distortion value has been obtained.

The threshold for the distortion value T_1 in ith pass is set as $5 + i/3$ in accordance with experience. Furthermore, the spatial motion correlation test ranges used in the current algorithm are 3×3 and 5×5, respectively. Since there are three passes during specific-block-size ME, the larger test range is applied in the first two passes, while the smaller test range is used in the final pass. Meanwhile, T_2 is assigned a value of 0 for block sizes larger than 8×8 and a value of 1 for smaller blocks. Note that this threshold is deliberately assigned a stricter value for larger blocks to prevent them from converging prematurely.

C. MV Propagation Method

The MV propagation technique employed in the current scheme is designed to enhance the degree of motion correlation in large textureless regions of the frame. When a block converges, i.e., it has a high degree of spatial motion correlation and a low distortion, its MV is automatically offered to the neighboring unconverged blocks as a potential MV for their own motions. If the difference between the original distortion value of the neighboring block and the new distortion value of the block produced by accepting the propagated MV is

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000233

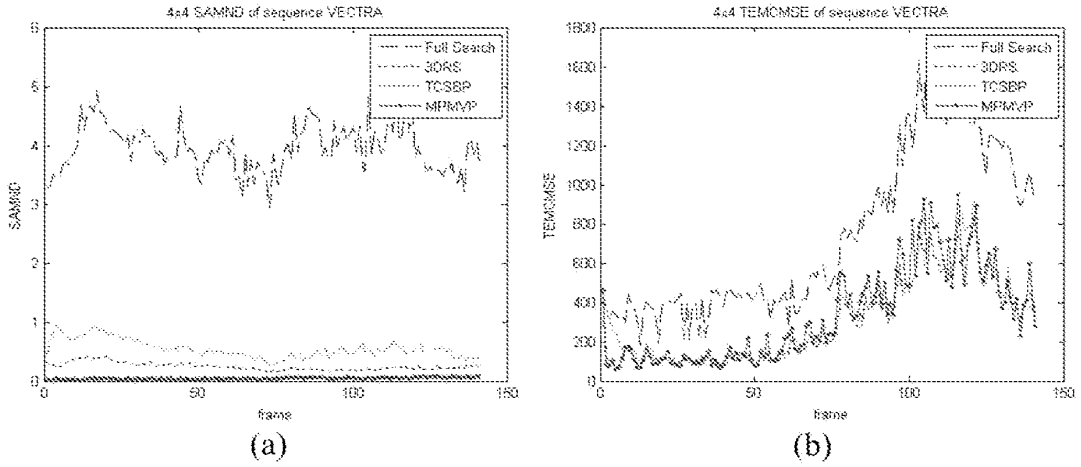


Fig. 10. SAMND and TEMC-MSE comparisons for full search (± 32), 3DRS, TCSBP and proposed algorithm. (Note results relate to VECTRA test sequence).

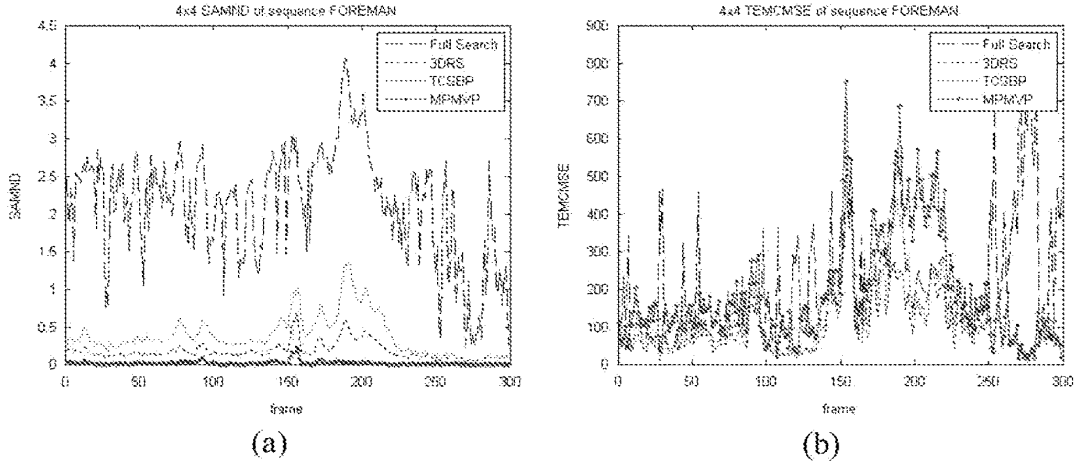


Fig. 11. SAMND and TEMC-MSE comparisons for full search (± 16), 3DRS, TCSBP and proposed algorithm. (Note results relate to FOREMAN test sequence).

sufficiently small, i.e., $15 - 2/3^{*}i$, the neighboring block accepts the propagated MV as its own MV. Having accepted the propagated MV, the block then checks to see whether or not the MV causes it to become converged. When a block has converged, irrespective of whether or not convergence is induced by the propagated MV, the same MV is propagated to the neighbors of the new converged block.

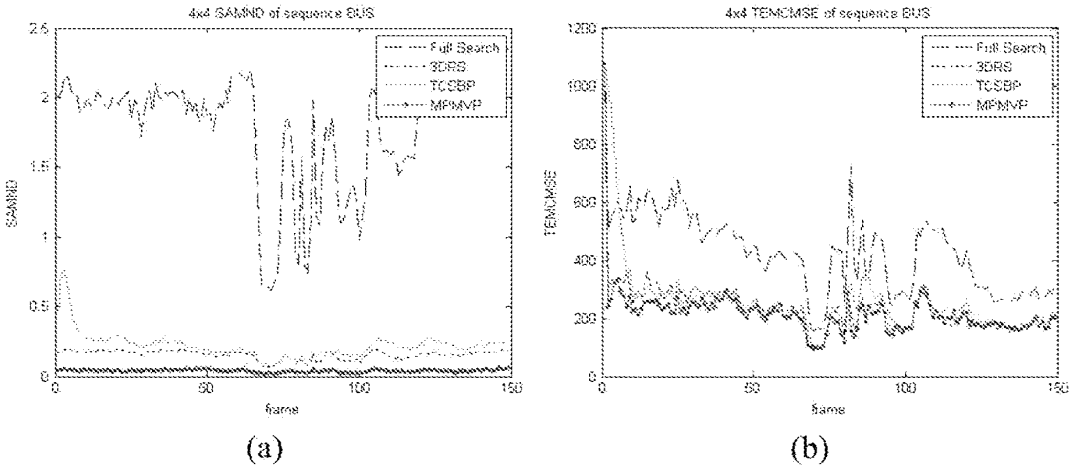
The convergence propagation technique is used to expand size of the region, which has the similar MV, into the size of object. Convergence propagation ensures that the MVs of the background regions of a frame have a high degree of spatial correlation. In other words, the background appears as a single, continuous region with no obvious areas of demarcation, even across strong edge areas, as shown in Fig. 6.

III. Experimental Results

In the figures discussed in the following, the MVFs and interpolated frames are based on a block size of 4×4. For example, a MV of block size 32×32 will be presented by 64 MVs of block size 4×4. Each MV is colored in red, originates from the center of the 4×4 block and terminates with a solid blue point. Fig. 7 presents the MVs of converged blocks obtained by the proposed ME scheme with different block sizes. The MVs of the background regions of the frame are found in the earlier passes of the ME search process performed using a larger block size. The use of a variable block size in the search procedure enables the block size of MV to gradually fit the contour of the objects in each pass and therefore the accuracy of the identified MVs is greatly enhanced. As a result, Fig. 7 demonstrates that the algorithm successfully divides the frame into two distinct objects, namely the background and the moving car.

In general, interpolated (i.e., up-converted) frames are generated using a motion compensation technique. Some uncovered regions (i.e., regions with no MVs) can be interpolated based on neighboring available pixels. Since the areas of these uncovered regions usually are small, we use spatial bilinear interpolation to interpolate uncovered regions in motion compensation step of this paper. Figs. 8 and 9 present the MVFs and corresponding interpolated frames obtained using the 3DRS [4], TCSBP [2] and proposed MPMVP ME algorithms for the BUS and VECTRA video sequences, respectively. Note that in every case, the test sequences are in a CIF format, have an original frame rate of 30 frames/s, and are up-converted to a rate of 60 frames/s. Observing these figures, it is apparent that the 3DRS and TCSBP

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000234



(a) (b)

Fig. 12. SAMND and TEMC-MSE comparisons for full search (±16), 3DRS, TCSBP and proposed algorithm. (Note results relate to BUS sequence).

schemes generate incorrect MVs in the smooth, expansive background regions of the frame because they fail to propagate the MVs efficiently among neighboring blocks.

The quality of the interpolated frames produced in FRUC applications is generally assessed using subjective visual quality measure. However, in [2], the authors proposed two new evaluation criteria with which to measure the motion correlation of interpolated frames. The first criterion was the spatial consistency indicator (i.e., the sum of absolute minimum neighboring difference, designated as SAMND), while the second criterion related to the temporal continuity of the motion field between successive frames (i.e., temporal extending motion compensation mean square error, designated as TEMC-MSE). SAMND evaluates the MV differences between neighboring blocks in accordance with

$$\mathrm{SAMND} = \sum_{\vec{X} \in \mathrm{MVF}_t} \min(\mathrm{abs}(\mathrm{Vec}_{\vec{X},t} - \mathrm{Vec}_{\vec{X}+\vec{\delta},t})) \quad (6)$$

where $\vec{X}$ is the block location, *Vec* is the MV of the block, and δ is the 3×3 neighboring region. Meanwhile, the TEMC-MSE criterion measures the temporal MV ductility and focuses on the smooth regions of the frame such as the background where the MVs in the previous frame are also valid for the current frame. The TEMC-MSE criterion has the form

$$\mathrm{TEMC-MSE} = \frac{\sum_{\vec{P}} (I(\vec{P},t) - I(\vec{P} - \mathrm{Vec}_{\vec{P},t-T}, t-T))^2}{M \times N} \quad (7)$$

where

I	luminance component of a frame;
$\vec{P}$	pixel position;
$M \times N$	total number of sampling pixels in the frame;
$\mathrm{Vec}_{\vec{P},t-T}$	MV of position $\vec{P}$ at frame $t-T$, and can be estimated from the luminance fields $I(\vec{P}, t-T)$ and $I(\vec{P}, t-2T)$.

Figs. 10–12 compares the SAMND and TEMC-MSE metrics for the full search, 3DRS, TCSBP and proposed MPMVP for the VECTRA sequence (characterized by fast and complicated motions), FOREMAN sequence (characterized by low motions) and BUS sequences (characterized by complicated motions), respectively. In general, Figs. 10–12 show that the MVs generated by the proposed algorithm have the highest degree of spatial correlation of all the MVs produced by the four algorithms. However, in Fig. 11(b), the proposed algorithm has the poorest TEMC-MSE performance. The poor performance arises because the proposed distortion criterion deliberately ignores small distortion values in order to enhance the noise resistance. However, when checking the up-converted FOREMAN sequence by human eyes, it is found that the proposed algorithm generates the highest quality interpolated frames of the four algorithms. For example, Fig. 13 shows frames 154 and 155 of the original FOREMAN sequence together with the corresponding interpolated frames generated using the full search, 3DRS, TSCBP and proposed MPMVP schemes, respectively. It can be seen the visual quality of MPMVP is the best. [Note that these particular frames are deliberately chosen here because frame 154 has a high TEMC-MSE value in Fig. 11(b)]. These figures also show that the full search scheme has the poorest performance of the four algorithms in terms of both the SAMND and the TEMC-MSE metrics since it is not specifically designed to focus on motion correlation.

Although the ME scheme proposed in this study is a recursive motion search method, the convergence decision rules and MV propagation technique greatly reduce its computational time and complexity. For full search, the computation is decided by search range, i.e., for search range n, the computation is $(2n+1)^2$ ADs/pixel, where n is often set as 7, 15, 31, or even 63. For a worst-case in MPMVP, the computation is 101 ADs/pixel for MPMVP. Because in the first pass, the proposed algorithm takes the motion vector in the previous field as the PMV and then evaluates the 13 search points defined in $\mathrm{US}_{\mathrm{large}}$ [see (3)]. In the following 11 passes, the algorithm evaluates 3 PMV predictors and each of the 5 search points defined in $\mathrm{US}_{\mathrm{small}}$. Therefore, the maximal computational complexity is 13+(11*(3+5)) = 101 ADs/pixel. Bus in general cases, the actually evaluated search points are much less than 101 ADs/pixel.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000235

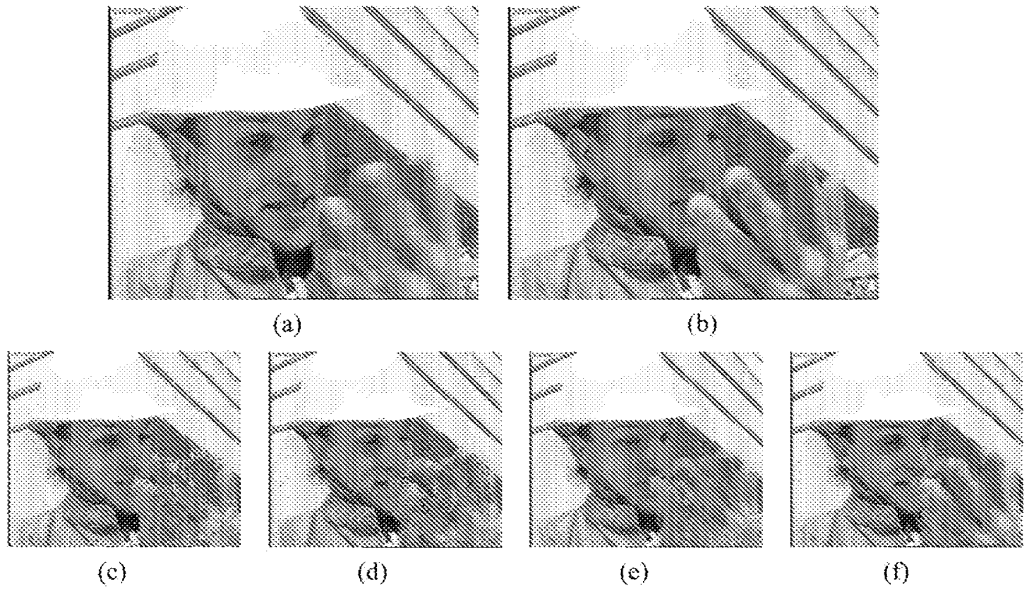


Fig. 13. (a), (b) Frames 154 and 155 in FOREMAN sequence. (c)-(f) Frames interpolated by full search, 3DRS, TCSBP and proposed algorithm, respectively.

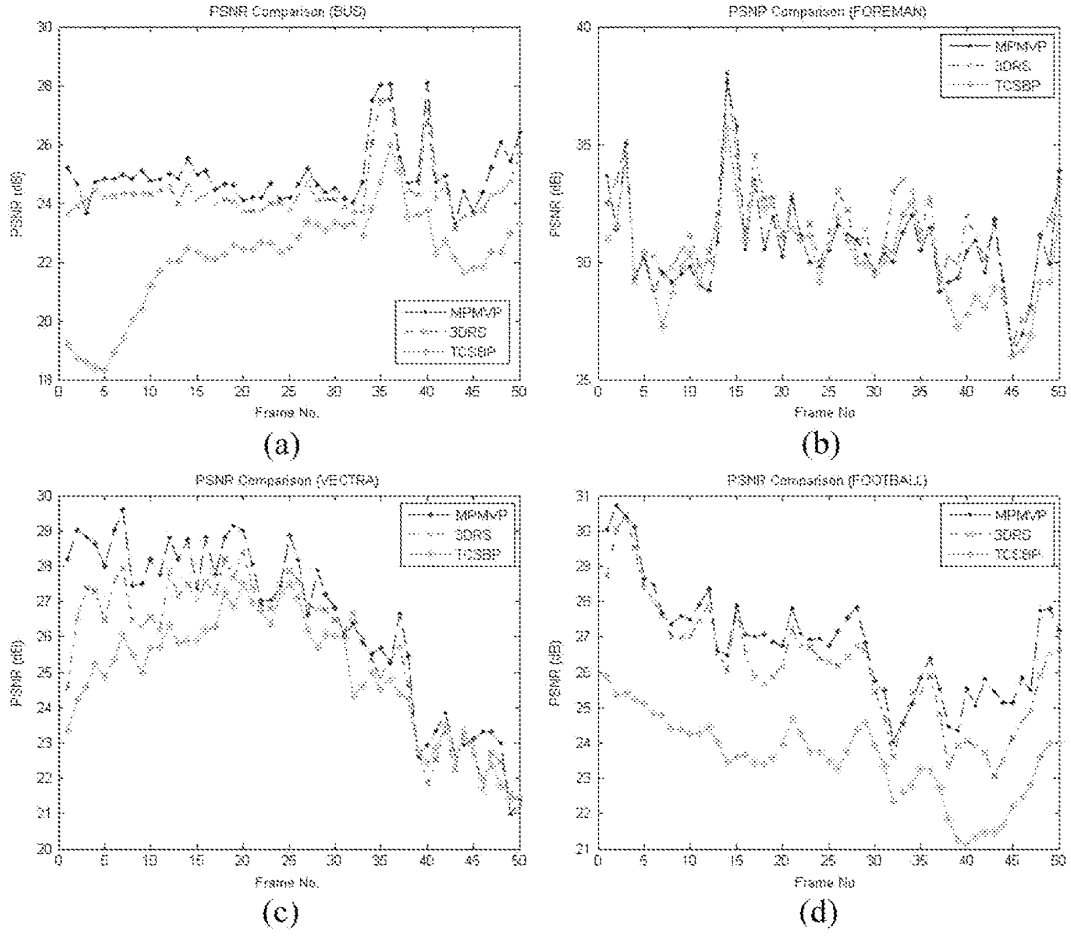


Fig. 14. PSNR comparisons on up-converted test sequences. (a) BUS. (b) FOREMAN. (c) VECTRA. (d) FOOTBALL. (a), (b), (c) are in CIF format and (d) is in de-interlaced D1 format.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000236

TABLE I
MOTION ESTIMATION TIMES SPENT IN 3DRS, TCSBP AND PROPOSED MPMVP ALGORITHM

Time (sec/frame) / Sequence	3DRS	TCSBP	MPMVP
BUS (CIF)	0.317	0.037	0.145
FOREMAN (CIF)	0.315	0.036	0.074
VECTRA (CIF)	0.305	0.038	0.116
FOOTBALL (D1)	2.186	0.264	1.143

We have tested the motion search time and PSNR performance for 3DRS, TCSBP and the proposed MPMVP scheme for three CIF format test sequences. The frame rate of these sequences are firstly down-sampled from 30 to 15 fps and then up-converted to 30 fps. We calculate the PSNR between the up-converted frames and the original frames in 30 fps sequence. The PSNR curves of the first 50 up-converted frames are shown in Fig. 14, while the corresponding motion estimation times are listed in Table I. In order to test the performance of the proposed MPMVP in larger resolution sequences, one de-interlaced D1 sequence (50 fps), FOOTBALL, is down-sampled from 50 to 25 fps and then up-converted to 50 fps. The experimental result of FOOTBALL is also shown in Fig. 14 and Table I. From those data, it can be seen that using our MPMVP scheme in frame rate up-conversion applications not only can achieve higher quality up-converted frames, but also can save significant computation.

IV. CONCLUSION

This paper has presented a new highly-accurate, low-complexity ME scheme for FRUC applications. The proposed algorithm is based upon a multi-pass, variable block-size approach and features a new noise-resistant distortion criterion to improve the quality of the interpolated frames. Furthermore, efficient MV convergence criteria and MV propagation techniques are employed to reduce the overall computational time and cost of the scheme. The experimental results have shown that the proposed ME scheme not only achieves a higher MV accuracy than existing ME schemes, but also yields significant computational savings. The quality of our true MV generated up-converted frames are better than other ME algorithm generated frames.

REFERENCES

[1] B.-T. Choi, S.-H. Lee, and S.-J. Ko, "New frame rate up-conversion using bi-directional motion estimation," *IEEE Trans. Consumer Electron.*, vol. 46, no. 3, pp. 603–609, Aug. 2000.

[2] J. Wang, D. Wang, and W. Zhang, "Temporal compensated motion estimation with simple block-based prediction," *IEEE Trans. Broadcast.*, vol. 49, no. 3, pp. 241–248, Sep. 2003.

[3] H. Taehyeun, L. Seongjoo, and K. Jaeseok, "Motion compensated frame interpolation by new block-based motion estimation algorithm," *IEEE Trans. Consumer Electron.*, vol. 50, no. 2, pp. 752–759, May 2004.

[4] G. de Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, "True-motion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 5, pp. 368–379, Oct. 1993.

[5] R. Braspenning and G. de Haan, "True-motion estimation using feature correspondence," *SPIE, Proc. VCIP*, pp. 396–407, Jan. 2004.

[6] G. de Haan and P. W. A. C. Biezen, "An efficient true-motion estimator using candidate vectors from a parametric motion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 1, pp. 85–91, Feb. 1998.

[7] A. Pelagotti and G. de Haan, "A new algorithm for high quality video format conversion," in *2001 Proc. Int. Conf. on Image Process.*, Oct. 2001, vol. 2, pp. 375–378.

[8] S.-H. Lee, Y.-C. Shin, Y. Seungjoon, H.-H. Moon, and R.-H. Park, "Adaptive motion-compensated interpolation for frame rate up-conversion," *IEEE Trans. Consumer Electron.*, vol. 48, no. 3, pp. 444–450, Aug. 2002.

[9] S.-H. Lee and S. Yang, "Adaptive motion-compensated frame rate up-conversion," *Electron. Lett.*, vol. 38, no. 10, pp. 451–452, May 2002.

[10] B.-W. Jeon, G.-I. Lee, S.-H. Lee, and R.-H. Park, "Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure," *IEEE Trans. Consumer Electron.*, vol. 49, no. 3, pp. 499–508, Aug. 2003.

[11] D. Wang and D. Lauzon, "Hybrid algorithm for estimating true motion fields," *Opt. Eng.*, vol. 39, no. 11, pp. 2876–2881, Nov. 2000.

[12] K. Hilman, H.-W. Park, and Y.-M. Kim, "Using motion compensated frame-rate conversion for the correction of 3:2 pulldown artifacts in video sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 869–877, Sep. 2000.

[13] P. Ishwar and P. Moulin, "On spatial adaptation of motion-field smoothness in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 980–989, Sep. 2000.

[14] S. Fujiwara and A. Taguchi, "Motion-compensated frame rate up-conversion based on block matching algorithm with multi-size blocks," in *Proc. 2005 Int. Symp. on Intell. Signal Process. Commun. Syst.*, Dec. 2005, vol. 13–16, pp. 353–356.

[15] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Frame rate up-conversion using perspective transform," *IEEE Trans. Consumer Electron.*, vol. 52, no. 3, Aug. 2006.

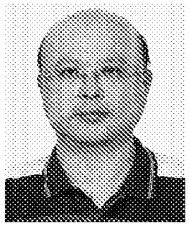
[16] O. EePing, H. Wang, and P. Xue, "Video coding based on true motion estimation," in *Proc. 2003 IEEE Int. Conf. on Acoust., Speech, and Signal Process.,*, Apr. 2003, vol. 3, no. 6–10, pp. 409–12.

[17] J. E. Santos Conde, A. Teuner, and B. J. Hosticka, "Hierarchical locally adaptive multigrid motion estimation for surveillance applications," in *Proc. 1999 IEEE Int. Conf. on Acoust., Speech, Signal Process.*, Mar. 1999, vol. 6, pp. 3365–3368.

[18] G. de Haan, "IC for motion-compensated de-interlacing, noise reduction, and picture-rate conversion," *IEEE Trans. Consumer Electron.*, vol. 45, no. 3, pp. 617–624, Aug. 1999.

[19] J.-W. Suh and Y.-S. Ho, "Error concealment technique based on optical flow," *Electron. Lett.*, vol. 38, no. 18, pp. 1020–1021, Aug. 2002.

[20] J.-Y. Pyun, J.-S. Lee, J.-W. Jeong, J.-H. Jeong, and S.-J. Ko, "Robust error concealment for visual communications in burst-packet-loss networks," *IEEE Trans. Consumer Electron.*, vol. 49, no. 4, pp. 1013–1019, Nov. 2003.



Shen-Chuan Tai received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1982 and 1986, respectively, and received the Ph.D. degree in computer science from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1989.

He is currently a professor in the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. His teaching and research interests include data compression, DSP, VLSI array processors, computerized electrocardiogram processing, and multimedia systems.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000237



Ying-Ru Chen received the B.S. degree in education from the Department of Information and Computer Education, National Taiwan Normal University, Taipei, Taiwan, R.O.C., in 2000, and the M.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, R.O.C., in 2002, and where she is currently working toward the Ph.D. degree candidate in the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C.

Her research interests include video signal processing, data compression, multimedia systems, and digital watermarking.



Zheng-Bin Huang received the B.S. degree in electrical engineering and the B.S. degree in mechanical engineering from National Cheng Kung University, Tainan, Taiwan, R.O.C., both in 2004, and the M.S. degree in electrical engineering from the National Cheng Kung University, Tainan, Taiwan, R.O.C., in 2006.

His research interests include video compression, image processing, and fast true motion estimation algorithms.



Chuen-Ching Wang received B.S., M.S., and Ph.D. degrees in electrical engineering from the National Cheng Kung University, Taiwan, R.O.C., in 1986, 1994, and 2003, respectively.

He was a section chief at Directorate General Telecommunications in the Ministry of Transportation and Communications. He is currently an assistant professor at the institute of electrical engineering, National Changhua University of Education, Taiwan, R.O.C. His research interests include digital signal processing, image processing, computer network, mobile computing and multimedia communications.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000238

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13666680	BICI ET AL.
	Examiner	Art Unit
	CLIFFORD HILAIRE	2488

CPC- SEARCHED		
Symbol	Date	Examiner
H04N19/52; H04N19/513; H04N19/176; H04N19/61; H04N19/51; H04N19/597; H04N19/56; H04N19/139; H04N19/573; H04N19/521; H04N19/103; H04N19/30	6/25/2015	CH

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
375	240	6/25/2015	CH

SEARCH NOTES		
Search Notes	Date	Examiner
See Attached EAST Search History Document	6/25/2015	CH
IEEEXplore Search	6/25/2015	CH

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

/CLIFFORD HILAIRE/ Examiner.Art Unit 2488	

To: thao.pham@nokia.com,sandy.fong-hou@nokia.com,jacqueline.linebarger@nokia.com
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 73658

Jul 06, 2015 05:31:56 AM

Dear PAIR Customer:

Nokia USA Inc.
Attn: Intellectual Property Rights Docketing
200 South Mathilda Ave
Sunnyvale, CA 94086
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 73658 , 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.
13666680	CTNF	07/06/2015	NC77198US-NP
	892	07/06/2015	NC77198US-NP
	1449	07/06/2015	NC77198US-NP

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

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82A (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
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.

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	13/666,680
Filing Date	11-01-2012
First Named Inventor	Mehmet Oguz BICI
Title	METHOD FOR CODING AND AN APPARATUS
Art Unit	2488
Examiner Name	HILAIRE, CLIFFORD
Attorney Docket Number	042933/467264

SIGNATURE of Applicant or Patent Practitioner

Signature	/Guy R. Gosnell/	Date (Optional)	9/9/2015
Name	Guy R. Gosnell	Registration Number	34,610
Title (if Applicant is a juristic entity)	Patent Practitioner		
Applicant Name (if Applicant is a juristic entity)	Nokia Technologies Oy		

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.

☐ *Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. 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 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.

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
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

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 10949

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

NOKIA TECHNOLOGIES OY

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	[signature]	Date (Optional)	23 January 2015
Name	Director, [illegible]		
Title	Legal and [illegible] Property		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☐ Total of forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. 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 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.

Jukka Nihtilä
Head, Business Development
Legal & IP

Electronic Acknowledgement Receipt

EFS ID:	23441894
Application Number:	13666680
International Application Number:	
Confirmation Number:	4782
Title of Invention:	METHOD FOR CODING AND AN APPARATUS
First Named Inventor/Applicant Name:	Mehmet Oguz BICI
Customer Number:	73658
Filer:	Guy Randall Gosnell/Kim Shaul
Filer Authorized By:	Guy Randall Gosnell
Attorney Docket Number:	NC77198-US-NP
Receipt Date:	09-SEP-2015
Filing Date:	01-NOV-2012
Time Stamp:	15:16:39
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	Miscellaneous Incoming Letter	467264Request.pdf	20500 2dccd2109905ebdab4e4597df533541a0f7088a2	no	1

Warnings:	
Information:	

2	Application Data Sheet	467264ADS.pdf	25395 84cb6ff32afa006fa7005a82d07f093bc8d39819	no	3
Warnings:					
Information:					
This is not an USPTO supplied ADS fillable form					
3	Assignee showing of ownership per 37 CFR 3.73	467264statement.pdf	86424 4e4d98087753bd84e5ac8d19b81401ce0db40934	no	3
Warnings:					
Information:					
4	Power of Attorney	467264POA.pdf	508180 7e639ecf8f0ac7d5dd08a8d7867bd94d76c92e9a	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	640499		

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.

Attorney's Docket No. 042933/467264 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re:	Mehmet Oguz Bici	Confirmation No.:	4782
Appl. No.:	13/666,680	Group Art No.:	2488
Filing Date:	November 1, 2012	Examiner:	Clifford Hilaire
Title:	METHOD FOR CODING AND AN APPARATUS		

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

Request to Correct or Update Applicant Name Under 37 CFR § 1.46(c)

This is a request to correct or update the original naming of Applicant(s) for the above-identified application.

☒ Please correct or update the name of the Applicant(s) to:

Applicant No. 1: Nokia Technologies Oy

☒ A Power of Attorney and 3.73(c) statement from each corrected or updated Applicant accompanies this Request.

☒ An Application Data Sheet with an updated Applicant Information Section accompanies this Request.

Applicant(s) respectfully requests that an updated Filing Receipt confirming these changes be issued.

Respectfully submitted,

/Guy R. Gosnell/

Guy R. Gosnell
Registration No. 34,610

CUSTOMER NO. 10949
ALSTON & BIRD LLP
Bank of America Plaza
101 South Tryon Street, Suite 4000
Charlotte, NC 28280-4000
Tel Charlotte Office (704) 444-1000
Fax Charlotte Office (704) 444-1111

Application Data Sheet

Application Information

Application number	13/666,680
Filing Date:	November 1, 2012
Title:	Method for Coding and an Apparatus
Attorney Docket Number:	042933/467264

Applicant Information

Applicant 1

Assignee: **Nokia Technologies Oy**

Legal Representative under 35 USC 117:

Joint Inventor: ~~Mehmet Oguz Bici~~

Person to whom the inventor is obligated to assign:

Person who shows sufficient proprietary interest:

If applicant is the legal representative, indicate the Authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:

If the Applicant is an Organization check here: ☒

Mailing Address

Address 1:	~~Tammelan puistokatu 1-3 D 46~~	**Karaportti 3**
Address 2:		
City:	~~Tampere~~	**Espoo**
State/Province:		
Country:	~~Finland~~	**Finland**
Postal Code:	~~33500~~	**02610**
Phone Number:		

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000246

Fax Number:

Email Address:

Applicant 2

Assignee:

Legal Representative under 35 USC 117:

Joint Inventor: ~~Jani Lainema~~

Person to whom the inventor is obligated to assign:

Person who shows sufficient proprietary interest:

If applicant is the legal representative, indicate the Authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:

If the Applicant is an Organization check here: ☐

Mailing Address

Address 1: ~~Kisakentankatu 12 B 6~~

Address 2:

City: ~~Tampere~~

State/Province:

Country: ~~Finland~~

Postal Code: ~~33230~~

Phone Number:

Fax Number:

Email Address:

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000247

Applicant 3

Assignee:

Legal Representative under 35 USC 117:

Joint Inventor: ~~Kemal Ugur~~

Person to whom the inventor is obligated to assign:

Person who shows sufficient proprietary interest:

If applicant is the legal representative, indicate the Authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:

If the Applicant is an Organization check here: ☐

Mailing Address

Address 1: ~~Lapintie 6D 25~~

Address 2:

City: ~~Tampere~~

State/Province:

Country: ~~Finland~~

Postal Code: ~~33100~~

Phone Number:

Fax Number:

Email Address:

Submitted by:

Signature /Guy R. Gosnell/ Date 2015-09-09

Printed Name Guy R. Gosnell Registration Number 34,610

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000248

PTO/AIA/96 (08-12)
Approved for use through 01/31/2013. 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.

STATEMENT UNDER 37 CFR 3.73(c)

Applicant/Patent Owner: Nokia Technologies Oy

Application No./Patent No.: 13/666,680 Filed/Issue Date: 11-01-2012

Titled: METHOD FOR CODING AND AN APPARATUS

Nokia Technologies Oy, a corporation

(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that, for the patent application/patent identified above, it is (choose **one** of options 1, 2, 3 or 4 below):

1. ☑ The assignee of the entire right, title, and interest.

2. ☐ An assignee of less than the entire right, title, and interest (check applicable box):

   ☐ The extent (by percentage) of its ownership interest is _____%. Additional Statement(s) by the owners holding the balance of the interest must be submitted to account for 100% of the ownership interest.

   ☐ There are unspecified percentages of ownership. The other parties, including inventors, who together own the entire right, title and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

3. ☐ The assignee of an undivided interest in the entirety (a complete assignment from one of the joint inventors was made). The other parties, including inventors, who together own the entire right, title, and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

4. ☐ The recipient, via a court proceeding or the like (*e.g.*, bankruptcy, probate), of an undivided interest in the entirety (a complete transfer of ownership interest was made). The certified document(s) showing the transfer is attached.

The interest identified in option 1, 2 or 3 above (not option 4) is evidenced by either (choose **one** of options A or B below):

A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

B. ☑ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Inventors To: Nokia Corporation

   The document was recorded in the United States Patent and Trademark Office at
   Reel 029555, Frame 0001, or for which a copy thereof is attached.

2. From: Nokia Corporation To: Nokia Technologies Oy

   The document was recorded in the United States Patent and Trademark Office at
   Reel 035258, Frame 0075, or for which a copy thereof is attached.

This collection of information is required by 37 CFR 3.73(b). 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.

PTO/AIA/96 (08-12)
Approved for use through 01/31/2013. 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.

STATEMENT UNDER 37 CFR 3.73(c)

3. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

4. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

5. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

6. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).

☑ As required by 37 CFR 3.73(c)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

/Guy R. Gosnell/	9/9/2015
Signature	Date
Guy R. Gosnell	34,610
Printed or Typed Name	Title or Registration Number

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.

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	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
13/666,680	11/01/2012	2488	2640	NC77198-US-NP	20	8

CONFIRMATION NO. 4782

REPLACEMENT FILING RECEIPT

10949
Nokia Corporation and Alston & Bird LLP
c/o Alston & Bird LLP
Bank of America Plaza, 101 South Tryon Street
Suite 4000
Charlotte, NC 28280-4000



Date Mailed: 09/17/2015

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Mehmet Oguz BICI, Tampere, TURKEY;
Jani LAINEMA, Tampere, FINLAND;
Kemal UGUR, Tampere, FINLAND;

Applicant(s)

Nokia Technologies Oy, Espoo, FINLAND;

Assignment For Published Patent Application

NOKIA CORPORATION, Espoo, FINLAND

Power of Attorney: The patent practitioners associated with Customer Number 10949

Domestic Priority data as claimed by applicant

This appln claims benefit of 61/555,703 11/04/2011

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

If Required, Foreign Filing License Granted: 11/20/2012

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 13/666,680**

Projected Publication Date: Not Applicable

Non-Publication Request: No

Early Publication Request: No

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000252

Title

METHOD FOR CODING AND AN APPARATUS

Preliminary Class

375

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications:

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000253

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

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
13/666,680	11/01/2012	Mehmet Oguz BICI	NC77198-US-NP

CONFIRMATION NO. 4782

10949
Nokia Corporation and Alston & Bird LLP
c/o Alston & Bird LLP
Bank of America Plaza, 101 South Tryon Street
Suite 4000
Charlotte, NC 28280-4000

POA ACCEPTANCE LETTER

OC000000077463222

Date Mailed: 09/17/2015

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 09/09/2015.

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.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/hsarwari/

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000255

To: usptomail@alston.com,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 10949

Sep 17, 2015 05:30:03 AM

Dear PAIR Customer:

Nokia Corporation and Alston & Bird LLP
c/o Alston & Bird LLP
Bank of America Plaza, 101 South Tryon Street
Suite 4000
Charlotte, NC 28280-4000
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 10949 , 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.
13666680	APP.FILE.REC	09/17/2015	NC77198-US-NP
	N570	09/17/2015	NC77198-US-NP

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

Attorney's Docket No. 042933/467264

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re:	Bici et al.	Confirmation No.:	4782
Appl. No.:	13/666,680	Group Art Unit:	2488
Filed:	November 1, 2012		
For:	METHOD FOR CODING AND AN APPARATUS		

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

REQUEST FOR CORRECTED FILING RECEIPT

The Application Data Sheet filed on November 1, 2012 contained errors in the Inventor Information section. Specifically, the residence provided for Inventor 1 is incorrect. A Subsequent Application Data Sheet is being filed concurrently herewith. Applicant requests that a corrected Filing Receipt be issued.

Respectfully submitted,

/Douglas E. McKay/

Douglas E. McKay
Registration No. 70,175

Customer No. 10949
ALSTON & BIRD LLP
Bank of America Plaza
101 South Tryon Street, Suite 4000
Charlotte, NC 28280-4000
Tel Charlotte Office (704) 444-1000
Fax Charlotte Office (704) 444-1111

Application Data Sheet

Application Information

Application number:	13/666,680
Filing Date:	November 1, 2012
Application Type:	Nonprovisional
Subject Matter:	Utility
Title:	METHOD FOR CODING AND AN APPARATUS
Attorney Docket Number:	042933/467264
Total Drawing Sheets:	13

Inventor Information

Inventor 1

Status:	Full Capacity
Given Name:	Mehmet
Middle Name:	Oguz
Family Name:	BICI
Non US Residency:	Yes
City of Residence:	Tampere
Country of Residence:	~~TR~~ **FI**
Street of Mailing Address:	Tammelan puistokatu 1-3 D 46
City of Mailing Address:	Tampere
Country of mailing address:	FI
Postal or Zip Code of mailing address:	33500

Inventor 2

Status:	Full Capacity
Given Name:	Jani
Middle Name:	
Family Name:	LAINEMA
Non US Residency:	Yes
City of Residence:	Tampere

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000258

Country of Residence:	FI
Street of Mailing Address:	Kisakentankatu 12 B 6
City of Mailing Address:	Tampere
Country of mailing address:	FI
Postal or Zip Code of mailing address:	33230

Inventor 3

Status:	Full Capacity
Given Name:	Kemal
Middle Name:	
Family Name:	UGUR
Non US Residency:	Yes
City of Residence:	Tampere
Country of Residence:	FI
Street of Mailing Address:	Lapintie 6D 25
City of Mailing Address:	Tampere
Country of mailing address:	FI
Postal or Zip Code of mailing address:	33100

Submitted by:

Signature /Douglas E. McKay/ Date 09/17/2015

Printed Name Douglas E. McKay Registration Number 70,175

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000259

Electronic Acknowledgement Receipt

EFS ID:	23533004
Application Number:	13666680
International Application Number:	
Confirmation Number:	4782
Title of Invention:	METHOD FOR CODING AND AN APPARATUS
First Named Inventor/Applicant Name:	Mehmet Oguz BICI
Customer Number:	10949
Filer:	Jason P. Cooper/Dennis Avery
Filer Authorized By:	Jason P. Cooper
Attorney Docket Number:	042933/467264
Receipt Date:	18-SEP-2015
Filing Date:	01-NOV-2012
Time Stamp:	11:14:08
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	Request for Corrected Filing Receipt	467264_Request_CFR_09-17-2015.pdf	90639 a561778649bc24d9c55fc4db0e007e76494434c2	no	1

Warnings:

Information:

2	Application Data Sheet	467264_Subsequent_ADS_09-17-2015.pdf	117748 42d38e4061da0c3c6ad9bfafeee3861c3800c2a3	no	2

Warnings:	
Information:	
This is not an USPTO supplied ADS fillable form	
Total Files Size (in bytes):	208387

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	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
13/666,680	11/01/2012	2488	2640	042933/467264	20	8

CONFIRMATION NO. 4782

CORRECTED FILING RECEIPT

10949
Nokia Corporation and Alston & Bird LLP
c/o Alston & Bird LLP
Bank of America Plaza, 101 South Tryon Street
Suite 4000
Charlotte, NC 28280-4000

OC000000077598891

Date Mailed: 09/23/2015

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Mehmet Oguz BICI, Tampere, FINLAND;
Jani LAINEMA, Tampere, FINLAND;
Kemal UGUR, Tampere, FINLAND;

Applicant(s)

Nokia Technologies Oy, Espoo, FINLAND;

Assignment For Published Patent Application

NOKIA CORPORATION, Espoo, FINLAND

Power of Attorney: The patent practitioners associated with Customer Number 10949

Domestic Priority data as claimed by applicant

This appln claims benefit of 61/555,703 11/04/2011

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

If Required, Foreign Filing License Granted: 11/20/2012

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 13/666,680**

Projected Publication Date: Not Applicable

Non-Publication Request: No

Early Publication Request: No

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000262

Title

METHOD FOR CODING AND AN APPARATUS

Preliminary Class

375

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications:

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000263

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

To: usptomail@alston.com,,
From: PAIR_eOfficeAction@uspto.gov
Cc: PAIR_eOfficeAction@uspto.gov
Subject: Private PAIR Correspondence Notification for Customer Number 10949

Sep 23, 2015 05:30:43 AM

Dear PAIR Customer:

Nokia Corporation and Alston & Bird LLP
c/o Alston & Bird LLP
Bank of America Plaza, 101 South Tryon Street
Suite 4000
Charlotte, NC 28280-4000
UNITED STATES

The following USPTO patent application(s) associated with your Customer Number, 10949 , 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.
13666680	APP.FILE.REC	09/23/2015	042933/467264

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

Substitute for form SB08 (Revised 07/09)				Complete if Known	
				Application Number	13/666,680
				Filing Date	11/01/2012
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				First Named Inventor	Mehmet Oguz Bici
				Art Unit	2488
(Use as many sheets as necessary)				Examiner Name	Clifford Hilaire
Sheet	*1*	of	1	Attorney Docket Number	042933/467264

OTHER DOCUMENTS

Examiner Initials*	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s) , volume-issue number(s), publisher, city and/or country where published.	English Language Translation Attached
	1	Office Action from corresponding Korean Patent Application No. 2014-7015093, dated August 21, 2015	YES
	2	**Wiegand, Tomas, et al.**; *"WD3: Working Draft 3 of High-Efficiency Video Coding"*; Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11; 5th Meeting; Geneva, CH, 16-23 March, 2011; Document JCTVC-E603; 239 pages	
	3	**Bross, Benjamin, et al.**; *"WD4: Working Draft 4 of High-Efficiency Video Coding"*; Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11; 6th Meeting; Torino, IT; 14-22 July 2011; Document JCTVC-F803_d; 232 pages	
	4	**J.-L. Lin, Y.-W. Chen, Y.-W. Huang, S. Lei;** *"CE9: Results of Experiment ROB04";* JCT-VC Doc. JCTVC-F052, Turin; Jul 2011	

Examiner Signature		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 with next communication to applicant.

CLT#35874174v1

Submitted September 25, 2015

Joint Collaborative Team on Video Coding (JCT-VC)
of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11
5th Meeting: Geneva, CH, 16-23 March, 2011

Document: JCTVC-E603

Title: **WD3: Working Draft 3 of High-Efficiency Video Coding**

Status: Output Document of JCT-VC

Purpose: Working Draft of HEVC

Author(s) or Contact(s):

Thomas Wiegand Fraunhofer HHI / TU Berlin	Email:	thomas.wiegand@hhi.fraunhofer.de
Woo-Jin Han Samsung Electronics	Email:	wjhan.han@samsung.com
Benjamin Bross Fraunhofer HHI	Email:	benjamin.bross@hhi.fraunhofer.de
Jens-Rainer Ohm RWTH Aachen	Email:	ohm@ient.rwth-aachen.de
Gary J. Sullivan Microsoft	Email:	garysull@microsoft.com

Source: Editor

Abstract

Working Draft 3 of High-Efficiency Video Coding.

Ed. Notes (WD3):

- Added Residual coding CABAC syntax and semantics
- Added Zig-zag scanning process
- Added CABAC Binarization processes
- Incorporated MV coding (JCTVC-E481)
- Incorporated Compression of reference indices (JCTVC-E059)
- Incorporated Zero merge candidate (JCTVC-E146)
- Incorporated Intra mode coding (JCTVC-E088/E131) (Inserted by TK 31/3/2011 with notes)
- Fixed the CABAC coefficients syntax, semantics and inverse scanning process
- Incorporated CABAC coeffs (JCTVC-E253)
- Moved the EGk binarization from the UEGk subclause in a separate subclause
- Added text representing CABAC entropy coding context initialisation
- Added text representing CABAC entropy coding context derivation.
- Mode-dependent 3- scan for intra (JCTVC-D393)
- Incorporated CABAC: Context size reduction (JCTVC-E227/E489)
- Incorporated CABAC: significance map coding simplification (JCTVC-E227/E338/E344/E494)
- Incorporated CABAC: Contexts for MVD (JCTVC-E324)
- Incorporated initial draft of CAVLC text
- CAVLC for 16x16 & 32x32 (JCTVC-E383)
- CAVLC table size reduction (JCTVC-E384)
- CAVLC for RQT (JCTVC-E404)
- CAVLC: counters (JCTVC-E143)
- CAVLC: Intra prediction mode coding in LCEC (JCTVC-D366)
- CAVLC: Inter prediction mode coding in LCEC (JCTVC- D370)
- CAVLC: 4x4 and 8x8 transform coefficient coding in LCEC (JCTVC- D374)
- Block-based ALF (JCTVC-E046/E323)
- ALF parameters to PPS (JCTVC-E045)
- Parallel deblocking (JCTVC-E496/E181/E224)
- Clipping for bi-pred averaging (JCTVC-E242)
- Reference sample padding (JCTVC-E488)
- Transformation processes are replaced by [TBD] mark (meeting note, JCTVC-E243)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000267

- Sub-LCU-level dQP (JCTVC-E051/E220)
- Temporal layer switching and reference list management based on temporal_id (JCTVC-E279/D081/D200)
- Improved text of entropy slice (JCTVC-D070) [Ed. (WJ): the term entropy slice is replaced by lightweight slice according to proponent's text]
- Slice independent deblocking and adaptive loop filtering (JCTVC-D128)
- Tiles, granularity slices (JCTVC-F483)
- PCM mode (JCTVC-E057 and JCTVC-E192)
- CAVLC: Inter pred coding (JCTVC-E381)
- CAVLC: Combined coding of inter prediction direction and reference frame index (JCTVC-D141)
- dQP (JCTVC-D125)
- Planar mode (JCTVC-E321)
- Luma-based chroma intra prediction (JCTVC-E266)
- Modification of DC predictor (JCTVC-E069)
- Bug in mapping table and the corresponding text for mostProbableIntra was fixed. (64x64 uses 3-directions, but the table was specified for 5-directions)
- Non-existing cases of Intra_DC, Intra_Planar and Intra_FromLuma are removed (due to reference sample padding, JCTVC-E488)

Ed. Notes (WD2):

- Incorporated Partial Merging according to JCTVC-D441
  - removed direct mode
  - moved merge to prediction_unit and added candidates
  - added partial merge restrictions
  - inter NxN partitioning only for smallest coding_unit
- Updated transform_tree and transform_coeff syntax
- Added transform_coeff to coding_unit syntax (Fix)
- Incorporated intra NxN partitioning only for smallest coding_unit according to JCTVC-D432
- Incorporated modified temporal motion vector predition according to JCTVC-D164
- Incorporated simplified motion vector prediction according to JCTVC-D231
  - removed median
  - removed pruning process
  - changed the selection manner of left/top predictor
- 8-tap luma interpolation filter according to JCTVC-D344
- 4-tap chroma interpolation filter according to JCTVC-D347
- Improved deblocking filter text according to JCTVC-D395
- IBDI syntax is removed
- Updated syntax and semantics
  - Two tool-enabling flags (adaptive_loop_filter_enabled_flag and cu_qp_delta_enabled_flag) are added in SPS according to software. However, low_delay_coding_enabled_flag is not added – it could be handled by more general reference management scheme. merging_enabled_flag is not added – partial merging (JCTVC-D441) was adopted thus merging cannot be turned off any more. amvp_mode[] is not added since amvp cannot be turned off any more due to absence of median predictor (JCTVC-D231). Note that software has all switches.
  - cu_qp_delta (coding unit layer), syntax and semantics are added. (JCTVC-D258)
  - collocated_from_l0 (slice header), syntax and semantics are added.
- Clean decoding refresh (CDR) (JCTVC-D234).
- Temporal motion vector memory compression (JCTVC-D072)
- Constrained intra prediction (JCTVC-D086)
- Mode-dependent intra smoothing (JCTVC-D282)
- Merging chroma intra prediction process into luma intra prediction process
- Combined reference list (JCTVC-D421)
- Chroma intra prediction mode reordering (JCTVC-D255/D278/D166)
- Adaptive loop filter text is added
- Entropy slice is added (JCTVC-D070)
- High precision bi-directional averaging (JCTVC-D321)
- Reduction of number of intra prediction modes for 64x64 blocks (JCTVC-D100)
- Misc.
  - TPE bits are reduced from 4 to 2
  - Clipping is applied to (temporally) scaled mv – revisit

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000268

Ed. Notes (WD1):

- Incorporated the decisions on high-level syntax according to JCTVC-B121
- Incorporated text from JCTVC-B205revision7
- Incorporated text from JCTVC-C319 (as found to be stable)
- Revised coding tree, coding unit and prediction unit syntaxes (coding tree syntax is newly added. needs to be confirmed)
- Initial drafting of decoding process of coding units in intra prediction mode (luma part, JCTVC-B100 and JCTVC-C042)
- Initial drafting of decoding process of coding units in inter prediction mode
- Initial drafting of scaling and transformation process
- Added text, transform 16T and 32T
- Initial drafting of deblocking process
- Improving the text, derivation process for motion vector components and reference indices
- Added text, boundary filtering strength

Open issues:

- Substantial portions of JCTVC-B205 have not been imported so far, as they require significant editorial work
- Should support for monochrome, 4:2:2 and 4:4:4 (with and w/o separate color planes) be included from the start? Currently, it has been left in the text as it doesn't seem to affect much text.
- Handling of the term "frame". We are currently leaning towards changing all occurrences of "frame" to "picture" (removed or marked all occurrences of "field")
- Large size table (de-scaling matrices) is not inserted yet.
- Slice-header syntaxes and their semantics are not completed yet. Also possible modifications that maybe necessary because of larger treeblocks (64x64) compared to macroblocks (16x16) are not yet considered.
- Text representing entropy coding named as "LCEC" are missing. The name LCEC needs to be adjusted as it propotes a property ("Low Complexity"). Currently, there is a place holder with CAVLC.
- Clipping is applied to temporally scaled motion vector in ~~8.4.2.1.7~~. Do we need this? (c.f. AVC seems not)
- SPS- and slice-level syntax items are significantly different from software in terms of both meaning and position. Needs to be unified.
- Use of bin string and bit string should be consistent.

~~- Luma-based chroma intra prediction (JCTVC-E266)~~
~~- Planar mode (JCTVC-E321)~~
~~- DC (JCTVC-E324)~~

- MD5 hash (JCTVC-E490)
- New additional loop filter (JCTVC-E049)

~~- Modification of DC predictor (JCTVC-E069)~~
~~- CAVLC: Inter pred coding (JCTVC-E381)~~
~~- CAVLC: Combined coding of inter prediction direction and reference frame index (JCTVC-D141)~~
~~- entropy slice – improved text (JCTVC-D070)~~
~~- slice independent deblocking and adaptive loop filtering (JCTVC-D328)~~

- ALF coefficient derivation from syntax elements
- SPS/PPS/slice-header syntax tables (with synchronization of software)

~~- PCM mode (JCTVC-F057 and JCTVC-E192)~~

- Improve text quality by considering: ~~Add CU, PU and TU definitions and replace treeblock with LCU~~strict use of terms "unit" and "block" – block = a rectangular 2D array (one component), unit = collective term for specifying information for both luma and chroma. Don't use ther term 'unit' by itself – always use the term 'unit' with prefix – coding unit, prediction unit or transform unit.
- Syntax functions cuAddress and granularity_block_boundary (fine granularity slices, JCTVC-F483) are missing
- Do we need SPS flag for disabling PCM mode? Currently PCM mode is turned on for all cases.
- Software seems to have SPS syntax to turn on/off combined reference list (JCTVC-D421), but currently text does not.

~~- CAVLC: Combined coding of inter prediction direction and reference frame index (JCTVC-D141) – corresponding software implementation seems to have more things (check of neighbouring reference indices) which is not described in JCTVC-D141, thus the relevent text was not included yet.~~

- Both variables IntraPredMode and IntraPredModeC are used in the syntax table but the actual derivations are specified in the decoding process. Maybe it's better to move them to the semantics section.
- Binarization of intra_chroma_pred_mode reflecting codeword switching and luma-based chroma intra prediction (JCTVC-E266) is missing.
- Clarification on the use of intra_chroma_pred_mode and IntraPredModeC is needed. The former specifies the syntax item to indicate how to determine the chroma intra prediction mode (IntraPredModeC) as 0 (to Intra_FromLuma), 1

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000269

(to Intra_Vertical), 2 (to Intra_Horizontal), 3 (Intra_DC or Intra_Planar) or 4 (re-use luma mode). The latter specifies chroma intra prediction mode, which is actually mapped to the specific prediction process.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000270

CONTENTS

Page

Abstract i

0 Introduction ..... 13~~11~~

0.1 Prologue ..... 13~~11~~

0.2 Purpose ..... 13~~11~~

0.3 Applications ..... 13~~11~~

0.4 Publication and versions of this Specification ..... 13~~11~~

0.5 Profiles and levels ..... 13~~11~~

0.6 Overview of the design characteristics ..... 14~~12~~

0.7 How to read this Specification ..... 14~~12~~

1 Scope ..... 15~~13~~

2 Normative references ..... 15~~13~~

3 Definitions ..... 16~~13~~

4 Abbreviations ..... 21~~18~~

5 Conventions ..... 22~~19~~

5.1 Arithmetic operators ..... 22~~19~~

5.2 Logical operators ..... 22~~19~~

5.3 Relational operators ..... 22~~20~~

5.4 Bit-wise operators ..... 23~~20~~

5.5 Assignment operators ..... 23~~20~~

5.6 Range notation ..... 23~~20~~

5.7 Mathematical functions ..... 23~~20~~

5.8 Order of operation precedence ..... 24~~21~~

5.9 Variables, syntax elements, and tables ..... 25~~22~~

5.10 Text description of logical operations ..... 26~~23~~

5.11 Processes ..... 27~~24~~

6 Source, coded, decoded and output data formats, scanning processes, and neighbouring relationships ..... 27~~24~~

6.1 Bitstream formats ..... 27~~24~~

6.2 Source, decoded, and output picture formats ..... 27~~24~~

6.3 Spatial subdivision of pictures and slices ..... 29~~26~~

6.4 Inverse scanning processes and derivation processes for neighbours ..... 30~~27~~

6.4.1 Inverse treeblock scanning process ..... 30~~27~~

6.5 Zig-zag scanning array initialisation process ..... 31~~28~~

6.6 Scanning order array initialisation process ..... 32~~28~~

7 Syntax and semantics ..... 32~~29~~

7.1 Method of specifying syntax in tabular form ..... 32~~29~~

7.2 Specification of syntax functions, categories, and descriptors ..... 33~~30~~

7.3 Syntax in tabular form ..... 35~~32~~

7.3.1 NAL unit syntax ..... 35~~32~~

7.3.2 Raw byte sequence payloads and RBSP trailing bits syntax ..... 36~~33~~

7.3.2.1 Sequence parameter set RBSP syntax ..... 36~~33~~

7.3.2.2 Picture parameter set RBSP syntax ..... 37~~33~~

7.3.2.3 Supplemental enhancement information RBSP syntax ..... 37~~34~~

7.3.2.4 Access unit delimiter RBSP syntax ..... 38~~34~~

7.3.2.5 Filler data RBSP syntax ..... 38~~34~~

7.3.2.6 Slice layer RBSP syntax ..... 38~~35~~

7.3.2.7 RBSP slice trailing bits syntax ..... 39~~35~~

7.3.2.8 RBSP trailing bits syntax ..... 39~~35~~

7.3.3 Slice header syntax ..... 40~~36~~

7.3.3.1 Reference picture list modification syntax ..... 41~~37~~

7.3.3.2 Reference picture lists combination syntax ..... 41~~37~~

7.3.3.3 Decoded reference picture marking syntax ..... 42~~38~~

7.3.3.4 Adaptive loop filter parameter syntax ..... 42~~39~~

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000271

7.3.4 Slice data syntax 45
7.3.5 Coding tree syntax 46
7.3.6 Coding unit syntax 47
7.3.7 Prediction unit syntax 48
7.3.8 Transform tree syntax 50
7.3.9 Transform coefficient syntax 52
7.3.10 Residual coding CABAC syntax 53
7.3.11 Residual coding CAVLC syntax 53
7.3.12 Level and sign CAVLC syntax 54
7.4 Semantics 55
7.4.1 NAL unit semantics 55
7.4.1.1 Encapsulation of an SODB within an RBSP (informative) 57
7.4.1.2 Order of NAL units and association to coded pictures, access units, and video sequences 58
7.4.2 Raw byte sequence payloads and RBSP trailing bits semantics 62
7.4.2.1 Sequence parameter set RBSP semantics 62
7.4.2.2 Picture parameter set RBSP semantics 64
7.4.2.3 Supplemental enhancement information RBSP semantics 65
7.4.2.4 Access unit delimiter RBSP semantics 65
7.4.2.5 Filler data RBSP semantics 65
7.4.2.6 Slice layer RBSP semantics 66
7.4.2.7 RBSP slice trailing bits semantics 66
7.4.2.8 RBSP trailing bits semantics 66
7.4.3 Slice header semantics 66
7.4.3.1 Reference picture list modification semantics 69
7.4.3.2 Reference picture lists combination semantics 70
7.4.3.3 Decoded reference picture marking semantics 70
7.4.3.4 Adaptive loop filter parameter semantics 72
7.4.4 Slice data semantics 74
7.4.5 Coding tree semantics 74
7.4.6 Coding unit semantics 75
7.4.7 Prediction unit semantics 76
7.4.8 Transform tree semantics 80
7.4.9 Transform coefficient semantics 81
7.4.10 Residual coding CABAC semantics 82
7.4.11 Residual coding CAVLC semantics 83
8 Decoding process 84
8.1 NAL unit decoding process 85
8.2 Slice decoding process 85
8.2.1 Decoding process for picture order count 85
8.2.1.1 Decoding process for picture order count type 0 85
8.2.1.2 Decoding process for picture order count type 1 85
8.2.1.3 Decoding process for picture order count type 2 85
8.2.2 Decoding process for reference picture lists construction 85
8.2.2.1 Decoding process for picture numbers 86
8.2.2.2 Initialisation process for reference picture lists 86
8.2.2.3 Modification process for reference picture lists 88
8.2.2.4 Mapping process for reference picture lists combination in B slices 89
8.2.3 Decoded reference picture marking process 90
8.2.3.1 Sequence of operations for decoded reference picture marking process 91
8.2.3.2 Decoding process for gaps in frame_num 91
8.2.3.3 Sliding window decoded reference picture marking process 91
8.2.3.4 Adaptive memory control decoded reference picture marking process 91
8.3 Decoding process for coding units coded in intra prediction mode 93
8.3.1 Derivation process for luma intra prediction mode ..... 오류! 책갈피가 정의되어 있지 않습니다. 84
8.3.2 Derivation process for chroma intra prediction mode 97
8.3.3 Decoding process for intra blocks 98
8.3.3.1 Intra sample prediction 99
8.4 Decoding process for coding units coded in inter prediction mode 109
8.4.1 Inter prediction process 110
8.4.2 Decoding process for prediction units in inter prediction mode 111
8.4.2.1 Derivation process for motion vector components and reference indices 112
8.4.2.2 Decoding process for inter prediction samples 123

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000272

8.4.3 Decoding process for the residual signal of coding units coded in inter prediction mode 130114
8.4.3.1 Decoding process for luma residual blocks 131114
8.4.3.2 Decoding process for chroma residual blocks 132115
8.5 Scaling, transformation and array construction process prior to deblocking filter process 133116
8.5.1 Scaling and transformation process 133116
8.5.2 Inverse scanning process for transform coefficients 134117
8.5.3 Scaling process for transform coefficients 134117
8.5.4 Transformation process for scaled transform coefficients 134118
8.5.4.1 Transformation process for 4 samples 136118
8.5.4.2 Transformation process for 8 samples 136118
8.5.4.3 Transformation process for 16 samples 136118
8.5.4.4 Transformation process for 32 samples 136121
8.5.5 Array construction process 136128
8.6 In-loop filter process 137128
8.6.1 Deblocking filter process 137128
8.6.1.1 Derivation process of transform unit boundary 138129
8.6.1.2 Derivation process of prediction unit boundary 138129
8.6.1.3 Derivation process of boundary filtering strength 139130
8.6.1.4 Filtering process for coding unit 140131
8.6.2 Adaptive loop filter process 148136
8.6.2.1 Derivation process for filter coefficients 150136
8.6.2.2 Derivation process for filter index array for luma samples 151137
8.6.2.3 Filtering process for luma samples 152138
8.6.2.4 Filtering process for chroma samples 153139
9 Parsing process 154139
9.1 Parsing process for Exp-Golomb codes 154140
9.1.1 Mapping process for signed Exp-Golomb codes 156141
9.2 CAVLC parsing process for slice data 156142
9.2.1 Parsing process for VLC codes 156142
9.2.2 Initialisation process 158144
9.2.2.1 Counter initialisation for codeword index adaptive mapping 159144
9.2.2.2 Initialisation process for lastPosVlcNumIndex 159145
9.2.2.3 Initialisation process for lastPosTable 159145
9.2.2.4 Initialisation process for splitPredPartModeTable 159145
9.2.2.5 Initialisation process for intraModeTable 159145
9.2.2.6 Initialisation process for cbpSplitTransTable 160146
9.2.3 Codeword index mapping update process 160146
9.2.4 Parsing process for transform coefficient syntax elements 161147
9.2.4.1 Parsing process for last_pos_level_one 161147
9.2.4.2 Parsing process for level_minus2_and_sign 162148
9.2.4.3 Parsing process for run_level_one 162148
9.2.4.4 Parsing process for level 166152
9.2.4.5 Parsing process for cu_split_pred_part_mode 166152
9.2.4.6 Parsing process for rem_intra_luma_pred_mode 167153
9.2.4.7 Parsing process for cbp_and_split_transform 167153
9.3 CABAC parsing process for slice data 169154
9.3.1 Initialisation process 169155
9.3.1.1 Initialisation process for context variables 169155
9.3.1.2 Initialisation process for the arithmetic decoding engine 181165
9.3.2 Binarization process 182166
9.3.2.1 Unary (U) binarization process 187169
9.3.2.2 Truncated unary (TU) binarization process 187170
9.3.2.3 Truncated Rice (TR) binarization process 188170
9.3.2.4 k-th order Exp-Golomb (EGk) binarization process 188170
9.3.2.5 Concatenated unary/ k-th order Exp-Golomb (UEGk) binarization process 188171
9.3.2.6 Fixed-length (FL) binarization process 189171
9.3.2.7 Binarization process for cu_qp_delta 189171
9.3.2.8 Binarization process for pred_type 189171
9.3.2.9 Binarization process for rem_intra_luma_pred_mode 190172
9.3.2.10 Binarization process for coeff_abs_level_minus3 190172
9.3.3 Decoding process flow 191173
9.3.3.1 Derivation process for ctxIdx 191172

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000273

9.3.3.2 Arithmetic decoding process 202183
9.3.4 Arithmetic encoding process (informative) 208189
9.3.4.1 Initialisation process for the arithmetic encoding engine (informative) 208189
9.3.4.2 Encoding process for a binary decision (informative) 209190
9.3.4.3 Renormalization process in the arithmetic encoding engine (informative) 210191
9.3.4.4 Bypass encoding process for binary decisions (informative) 212193
9.3.4.5 Encoding process for a binary decision before termination (informative) 213194
9.3.4.6 Byte stuffing process (informative) 215196
Annex A Profiles and levels 216197
A.1 Requirements on video decoder capability 216197
A.2 Profiles 216197
A.2.1 ABC profile 216197
A.3 Levels 216197
Annex B Byte stream format 218199
B.1 Byte stream NAL unit syntax and semantics 218199
B.1.1 Byte stream NAL unit syntax 218199
B.1.2 Byte stream NAL unit semantics 218199
B.2 Byte stream NAL unit decoding process 219200
B.3 Decoder byte-alignment recovery (informative) 219200
Annex C Hypothetical reference decoder 220201
C.1 Operation of coded picture buffer (CPB) 221202
C.1.1 Timing of bitstream arrival 221202
C.1.2 Timing of coded picture removal 221203
C.2 Operation of the decoded picture buffer (DPB) 221202
C.3 Bitstream conformance 221203
C.4 Decoder conformance 221202
Annex D Supplemental enhancement information 222203
D.1 SEI payload syntax 222203
D.2 SEI payload semantics 222203
Annex E Video usability information 223204
E.1 VUI syntax 223204
E.1.1 VUI parameters syntax 223204
E.2 VUI semantics 223204

LIST OF FIGURES

Figure 6-2 – Nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a picture [Ed. Re-draw figure] 2926
Figure 6-3 – Nominal vertical and horizontal locations of 4:4:4 luma and chroma samples in a picture [Ed. Re-draw figure] 2926
Figure 6-4 – A picture with 11 by 9 treeblocks that is partitioned into two slices 3027
Figure 7-1 – Structure of an access unit not containing any NAL units with nal_unit_type equal to 0, 7, 8, or in the range of 12 to 18, inclusive, or in the range of 20 to 31, inclusive 6156
Figure 8-2 – Intra prediction angle definition (informative) 10592
Figure 8-3 – Spatial neighbours that can be used as merging candidates (informative) illustrates the position of the spatial neighbours A, B, C and D relative to the current prediction unit. 11699
Figure 8-4 – Spatial neighbours, that can be used to derive reference indices for temporal merging candidates, (informative) illustrates the position of the spatial neighbours A, B, C, D and E relative to the current prediction unit. 117101
Figure 8-5 – Spatial motion vector neighbours 119103
Figure 8-6 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation 126109
Figure 8-8 Mapping between geometric position and luma adaptive loop filter index according to alfTap (informative) 153139

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000274

Figure 9-1 – Overview of the arithmetic decoding process for a single bin (informative) 202183
Figure 9-2 – Flowchart for decoding a decision 204185
Figure 9-3 – Flowchart of renormalization 206187
Figure 9-4 – Flowchart of bypass decoding process 207188
Figure 9-5 – Flowchart of decoding a decision before termination 208189
Figure 9-6 – Flowchart for encoding a decision 210191
Figure 9-7 – Flowchart of renormalization in the encoder 211192
Figure 9-8 – Flowchart of PutBit(B) 212193
Figure 9-9 – Flowchart of encoding bypass 213194
Figure 9-10 – Flowchart of encoding a decision before termination 214195
Figure 9-11 – Flowchart of flushing at termination 215196
Figure C-9-12 – Structure of byte streams and NAL unit streams for HRD conformance checks 220201
Figure C-9-13 – HRD buffer model 221202

LIST OF TABLES

Table 5-1 – Operation precedence from highest (at top of table) to lowest (at bottom of table) 2523
Table 6-1 – SubWidthC, and SubHeightC values derived from chroma_format_idc and separate_colour_plane_flag 2826
Table 7-1 – NAL unit type codes, syntax element categories, and NAL unit type classes 5652
Table 7-2 – Meaning of primary_pic_type 6560
Table 7-3 – Name association to slice_type 6664
Table 7-4 – modification_of_pic_nums_idc operations for modification of reference picture lists 6964
Table 7-5 – Interpretation of adaptive_ref_pic_marking_mode_flag 7165
Table 7-6 – Memory management control operation (memory_management_control_operation) values 7266
Table 7-7 – Specification of alf_chroma_idc 7468
Table 7-8 – Specification of cu_split_pred_part_mode when log2CUSize is greater than Log2MinCUSize 7569
Table 7-9 – Specification of cu_split_pred_part_mode when log2CUSize is equal to Log2MinCUSize 7569
Table 7-10 - Name association to prediction mode and partitioning type 7670
Table 7-11 – Number of bins and number of mode for rem_intra_luma_pred_mode associated to PuSize 7770
Table 7-12 – Name association to inter prediction mode 7871
Table 7-13 – Specification of cbp_and_split_transform 8173
Table 7-14 – Specification of ScanType[log2TrafoSize – 2][IntraPredMode] 8274
Table 7-15 – Specification of ScanTypeC[log2TrafoSize – 2][IntraPredModeC] 8274
Table 8-1 – Specification of intraPredModeNum 9485
Table 8-2 – Specification of mapIntraPredMode5 and mapIntraPredMode9 9586
Table 8-3 – Specification of IntraPredModeC according to the values of intra_chroma_pred_mode and IntraPredMode[xB][yB] 9887
Table 8-4 – Specification of intraPredMode and associated names 10089
Table 8-5 – Specification of intraFilterType[nS][IntraPredMode] for various prediction unit sizes 10290
Table 8-6 – Specification of intraPredOrder 10592
Table 8-7 – Specification of intraPredAngle 10593

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000275

Table 8-8 – Specification of invAngle ..... 106~~93~~
Table 8-9 – Assignment of the luma prediction sample predSampleLX$_L$[x_L, y_L] ..... 127~~112~~
Table 8-10 – Assignment of the chroma prediction sample predSampleLX$_C$[x_C, y_C] for (X, Y) being replaced by (1, b), (2, c), (3, d), (4, e), (5, f), (6, g), and (7, h), respectively ..... 130~~111~~
Table 8-11 – Derivation of threshold variables β and t_C from input Q ..... 147~~136~~
Table 8-12 – Specification of horPos[i] according to alfTap for adaptive loop filter process of luma samples ..... 152~~139~~
Table 8-13 – Specification of verPos[i] according to alfTap for adaptive loop filter process of luma samples ..... 153~~139~~
Table 9-1 – Bit strings with "prefix" and "suffix" bits and assignment to codeNum ranges (informative) ..... 155~~141~~
Table 9-2 – Exp-Golomb bit strings and codeNum in explicit form and used as ue(v) (informative) ..... 155~~142~~
Table 9-3 – Assignment of syntax element to codeNum for signed Exp-Golomb coded syntax elements se(v) ..... 156~~143~~
Table 9-4 – codeNum and codeword with vlcNum equal to 15 ..... 158~~145~~
Table 9-5 – Specification of initial values of lastPosTable[tableNum][codeNum] ..... 159~~146~~
Table 9-6 – Specification of intraModeTable[k][codeNum] ..... 160~~146~~
Table 9-7 – Specification of initial values of cbpSplitTransTable[flagPattern-8][n][codeNum] ..... 160~~147~~
Table 9-8 – Specification of lastPosVlcNumTable[blockType][vlcNumIdx] ..... 162~~149~~
Table 9-9 – Derivation of vlcNum from tableIdx and maxRunIdx ..... 164~~150~~
Table 9-10 – Derivation of largeOnePos from tableIdx and maxRunIdx when N is equal to 4 ..... 164~~151~~
Table 9-11 – Derivation of largeOnePos from tableIdx and maxRunIdx when N is greater than 4 ..... 164~~151~~
Table 9-12 – Derivation of runOfZeros from maxRun and codeNum ..... 165~~151~~
Table 9-13 – Derivation of levelThreshold from vlcIdx ..... 166~~153~~
Table 9-14 – Specification of cbpVlcNumTable[cbpVlcNumIdx] ..... 168~~155~~
Table 9-15 – Derivation of cMax from flagPattern and n ..... 168~~155~~
Table 9-16 – Specification of cbpSplitTransTable and counterNum associated ..... 168~~155~~
Table 9-17 – Association of ctxIdx and syntax elements for each slice type in the initialisation process ..... 170~~157~~
Table 9-18 – Values of variables m and n for alf_cu_flag ctxIdx ..... 172~~157~~
Table 9-19 – Values of variables m and n for split_coding_unit_flag ctxIdx ..... 172~~158~~
Table 9-20 – Values of variables m and n for skip_flag ctxIdx ..... 172~~158~~
Table 9-21 – Values of variables m and n for cu_qp_delta ctxIdx ..... 172~~158~~
Table 9-22 – Values of variables m and n for pred_type ..... 173~~158~~
Table 9-23 – Values of variable m and n for prev_intra_luma_pred_flag ctxIdx ..... 173~~158~~
Table 9-24 – Values of variable m and n for rem_intra_luma_pred_mode ctxIdx ..... 173~~158~~
Table 9-25 – Values of variable m and n for intra_chroma_pred_mode ctxIdx ..... 173~~159~~
Table 9-26 – Values of variable m and n for merge_flag ctxIdx ..... 173~~159~~
Table 9-27 – Values of variable m and n for merge_idx ctxIdx ..... 173~~159~~
Table 9-28 – Values of variable m and n for inter_pred_flag ctxIdx ..... 174~~159~~
Table 9-29 – Values of variable m and n for ref_idx_lc, ref_idx_l0, ref_idx_l1 ctxIdx ..... 174~~159~~
Table 9-30 – Values of variable m and n for mvd_lc, mvd_l0, mvd_l1 ctxIdx ..... 174~~159~~
Table 9-31 – Values of variable m and n for mvp_idx_lc, mvp_idx_l0, mvp_idx_l1 ctxIdx ..... 174~~160~~
Table 9-32 – Values of variable m and n for no_residual_data_flag ctxIdx ..... 174~~160~~
Table 9-33 – Values of variable m and n for split_transform_flag ctxIdx ..... 174~~160~~

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000276

Table 9-34 – Values of variable m and n for cbf_luma ctxIdx ..... 175~~160~~
Table 9-35 – Values of variable m and n for cbf_cb ctxIdx ..... 175~~160~~
Table 9-36 – Values of variable m and n for cbf_cr ctxIdx ..... 175~~160~~
Table 9-37 – Values of variable m and n for last_significant_coefficient_x ctxIdx ..... 176~~161~~
Table 9-38 – Values of variable m and n for last_significant_coefficient_y ctxIdx ..... 177~~162~~
Table 9-39 – Values of variable m and n for significant_coeff_flag ctxIdx (A) ..... 178~~163~~
Table 9-40 – Values of variable m and n for significant_coeff_flag ctxIdx (B) ..... 178~~163~~
Table 9-41 – Values of variable m and n for significant_coeff_flag ctxIdx (C) ..... 179~~164~~
Table 9-42 – Values of variable m and n for coeff_abs_level_greater1_flag ctxIdx ..... 180~~165~~
Table 9-43 – Values of variable m and n for coeff_abs_level_greater2_flag ctxIdx ..... 181~~166~~
Table 9-44 – Syntax elements and associated types of binarization, maxBinIdxCtx, ctxIdxTable, and ctxIdxOffset 183~~168~~
Table 9-45 – Bin string of the unary binarization (informative) ..... 187~~171~~
Table 9-46 – Binarization for pred_type ..... 190~~173~~
Table 9-47 – Binarization for rem_intra_luma_pred_mode ..... 190~~173~~
Table 9-48 – Specifcation of cTRMax depending on cRiceParam ..... 191~~174~~
Table 9-49 – Assignment of ctxIdxInc to binIdx for all ctxIdxTable and ctxIdxOffset values ..... 193~~176~~
Table 9-50 – Specifcation of ctxIdxInc using left and above syntax elements ..... 198~~179~~
Table 9-51 – Specifcation of lastCtx depending on binIdx ..... 199~~180~~
Table 9-52 – Specification of rangeTabLPS depending on pStateIdx and qCodIRangeIdx ..... 204~~186~~
Table 9-53 – State transition table ..... 205~~187~~
Table A-9-54 – Level limits [Ed. (TW) Kept for convenience] ..... 217~~199~~

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000277

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 standardising telecommunications on a world-wide basis. The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups that, 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 that fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

ISO (the International Organization for Standardization) 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 through 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 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.

This Recommendation | International Standard was prepared jointly by ITU-T SG 16 Q.6, also known as VCEG (Video Coding Experts Group), and by ISO/IEC JTC 1/SC 29/WG 11, also known as MPEG (Moving Picture Experts Group). VCEG was formed in 1997 to maintain prior ITU-T video coding standards and 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 digital 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.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000278

Rec. / IS

High-Efficiency Video Coding

0 Introduction

This clause does not form an integral part of this Recommendation | International Standard.

0.1 Prologue

This subclause does not form an integral part of this Recommendation | International Standard.

As the costs for both processing power and memory have reduced, network support for coded video data has diversified, and advances in video coding technology have progressed, the need has arisen for an industry standard for compressed video representation with substantially increased 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 Collaborative Team on Video Coding (JCT-VC) in 2010 for development of a new Recommendation | International Standard.

0.2 Purpose

This subclause does not form an integral part of this Recommendation | International Standard.

[Ed. TW: revise the following] 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 | International Standard 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 subclause does not form an integral part of this Recommendation | International Standard.

[Ed. TW: revise the following] 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 Publication and versions of this Specification

This subclause does not form an integral part of this Recommendation | International Standard.

This Specification has been jointly developed by ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group. It is published as technically-aligned twin text in both organizations ITU-T and ISO/IEC.

0.5 Profiles and levels

This subclause does not form an integral part of this Recommendation | International Standard.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000279

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 these have been integrated into a single syntax. Hence, this Specification will facilitate video data interchange among different applications.

Considering the practicality of implementing the full syntax of this Specification, however, a limited number of subsets of the syntax are also stipulated by means of "profiles" and "levels". These and other related terms are formally defined in clause 3.

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 syntax elements in the bitstream such as the specified size of the decoded pictures. In many applications, 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 values of the syntax elements in the bitstream. These constraints may be simple limits on values. Alternatively they may take the form of constraints on arithmetic combinations of values (e.g., picture width multiplied by picture height multiplied by number of pictures decoded per second).

Coded video content conforming to this Recommendation | International Standard uses a common syntax. In order to achieve a subset of the complete syntax, flags, parameters, and other syntax elements are included in the bitstream that signal the presence or absence of syntactic elements that occur later in the bitstream.

0.6 Overview of the design characteristics

This subclause does not form an integral part of this Recommendation | International Standard.

[Ed. TW: revise the following] The coded representation specified in the syntax is designed to enable a high compression capability for a desired image or video quality. The algorithm is typically 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. Encoding algorithms (not specified in this Recommendation | International Standard) may select between inter and intra coding for block-shaped regions of each picture. Inter coding uses motion vectors for block-based inter prediction to exploit temporal statistical dependencies between different pictures. Intra coding uses various spatial prediction modes to exploit spatial statistical dependencies in the source signal for a single picture. 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 visual 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 coding or arithmetic coding.

0.7 How to read this Specification

This subclause does not form an integral part of this Recommendation | International Standard.

0.8 It is suggested 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 6.67 (Scanning order array initialisation process

Input to this process is a block size blkSize.

Output of this process is the array ScanOrder[scanIdx][pos][comp]. The array index scanIdx equal to 0 specifies a zig-zag scan as specified in subclause 오류! 참조 원본을 찾을 수 없습니다., scanIdx equal to 1 specifies a horizontal scan and scanIdx equal to 2 specifies a vertical scan. The array index pos specifies the scan position ranging from 0 to (blkSize * blkSize) – 1. The array index comp equal to 0 specifies the horizontal component and the array index comp equal to 1 specifies the vertical component. The array ScanOrder is derived as follows.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000280

```
ScanOrder[0] = ZigZag

i = 0
y = 0
while( y < blkSize ) {
    x = 0
    while( x < blkSize ) {
        ScanOrder[ 1 ][ i ][ 0 ] = x
        ScanOrder[ 1 ][ i ][ 1 ] = y
        x++
        i++
    }
    y++
}

i = 0
x = 0
while( x < blkSize ) {
    y = 0
    while( y < blkSize ) {
        ScanOrder[ 2 ][ i ][ 0 ] = x
        ScanOrder[ 2 ][ i ][ 1 ] = y
        y++
        i++
    }
    x++
}
```

Syntax and semanticsSyntax and semantics) specifies the order to parse syntax elements from the bitstream. See subclauses 7.1-7.3 for syntactical order and see subclause 7.3.107.4 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 7.4.108 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. Annexes A through E also form an integral part of this Recommendation | International Standard.

Annex A specifies profiles each being tailored to certain application domains, and defines the so-called levels of the profiles. Annex B specifies syntax and semantics of a byte stream format for delivery of coded video as an ordered stream of bytes. Annex C specifies the hypothetical reference decoder and its use to check bitstream and decoder conformance. Annex D specifies syntax and semantics for supplemental enhancement information message payloads. Annex E specifies syntax and semantics of the video usability information parameters of the sequence parameter set.

Throughout this Specification, statements appearing with the preamble "NOTE -" are informative and are not an integral part of this Recommendation | International Standard.

1 Scope

This document specifies High-Efficiency 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. [Ed. TW: revise the following]

- ITU-T Recommendation T.35 (2000), *Procedure for the allocation of ITU-T defined codes for non-standard facilities*.
- ISO/IEC 11578:1996, Annex A, *Universal Unique Identifier*.
- ISO/CIE 10527:2007, *Colorimetric Observers*.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000281

3 Definitions

[Ed. (TW) adpated definitions so far. Needs more work including turning them into 1 sentence each.]

For the purposes of this Recommendation | International Standard, the following definitions apply:

3.1 **access unit**: A set of *NAL units* that are consecutive in *decoding order* and contain exactly one *primary coded picture*. In addition to the *primary coded picture* one *auxiliary coded picture*, or other *NAL units* not containing *slices* of a *primary coded picture*. The decoding of an access unit always results in a *decoded picture*.

3.2 **AC transform coefficient**: Any *transform coefficient* for which the *frequency index* in one or both dimensions is non-zero.

3.3 **adaptive binary arithmetic decoding process**: An entropy *decoding process* that derives the values of *bins* from a *bitstream* produced by an *adaptive binary arithmetic encoding process*.

3.4 **adaptive binary arithmetic encoding process**: An entropy *encoding process*, not normatively specified in this Recommendation | International Standard, that codes a sequence of *bins* and produces a *bitstream* that can be decoded using the *adaptive binary arithmetic decoding process*.

3.5 **B slice**: A *slice* that may be decoded using *intra prediction* or *inter prediction* using at most two *motion vectors* and *reference indices* to *predict* the sample values of each *block*.

3.6 **bin**: One bit of a *bin string*.

3.7 **binarization**: A set of *bin strings* for all possible values of a *syntax element*.

3.8 **binarization process**: A unique mapping process of all possible values of a *syntax element* onto a set of *bin strings*.

3.9 **bin string**: A string of *bins*. A bin string is an intermediate binary representation of values of *syntax elements* from the *binarization* of the *syntax element*.

3.10 **bi-predictive slice:** See *B slice*.

3.11 **bitstream**: A sequence of bits that forms the representation of *coded pictures* and associated data forming one or more *coded video sequences*. Bitstream is a collective term used to refer either to a *NAL unit stream* or a *byte stream*.

3.12 **block**: An MxN (M-column by N-row) array of samples, or an MxN array of *transform coefficients*.

3.13 **broken link**: A location in a *bitstream* at which it is indicated that some subsequent *pictures* in *decoding order* may contain serious visual artefacts due to unspecified operations performed in the generation of the *bitstream*.

3.14 **byte**: A sequence of 8 bits, written and read with the most significant bit on the left and the least significant bit on the right. When represented in a sequence of data bits, the most significant bit of a byte is first.

3.15 **byte-aligned**: A position in a *bitstream* is byte-aligned when the position is an integer multiple of 8 bits from the position of the first bit in the *bitstream*. A bit or *byte* or *syntax element* is said to be byte-aligned when the position at which it appears in a *bitstream* is byte-aligned.

3.16 **byte stream**: An encapsulation of a *NAL unit stream* containing *start code prefixes* and *NAL units* as specified in Annex B.

3.17 **can**: A term used to refer to behaviour that is allowed, but not necessarily required.

3.18 **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 Cb and Cr.

NOTE – 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.

3.19 **clean decoding refresh (CDR) access unit**: An *access unit* in which the *primary coded picture* is a *CDR picture*.

3.20 **clean decoding refresh (CDR) picture**: A *coded picture* containing only *slices* with *I slice types* that causes the *decoding process* to mark all *reference pictures* except the current CDR picture as "unused for reference" immediately before the decoding of the first picture following the current CDR with an *output order* greater than the current CDR picture. All *coded pictures* that follow a CDR picture in *output order* can be decoded without *inter prediction* from any *picture* that precedes the CDR picture in *output order*. All *coded pictures* with *output order* smaller than the current CDR are not affected by the deferred marking process.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000282

3.21 **coded picture**: A *coded representation* of a *picture*.

3.22 **coded picture buffer (CPB)**: A first-in first-out buffer containing *access units* in *decoding order* specified in the *hypothetical reference decoder* in Annex 0.

3.23 **coded representation**: A data element as represented in its coded form.

3.24 **coded slice NAL unit**: A *NAL unit* containing a *slice*.

3.25 **coded video sequence**: A sequence of *access units* that consists, in decoding order, of an *IDR access unit* followed by zero or more non-IDR *access units* including all subsequent *access units* up to but not including any subsequent *IDR access unit*.

3.26 **component**: An array or single sample from one of the three arrays (*luma* and two *chroma*) that make up a *picture* in 4:2:0, 4:2:2, or 4:4:4 colour format or the array or a single sample of the array that make up a *picture* in monochrome format.

3.27 **context variable**: A variable specified for the *adaptive binary arithmetic decoding process* of a *bin* by an equation containing recently decoded *bins*.

3.28 **DC transform coefficient**: A *transform coefficient* for which the *frequency index* is zero in all dimensions.

3.29 **decoded picture**: A *decoded picture* is derived by decoding a *coded picture*.

3.30 **decoded picture buffer (DPB)**: A buffer holding *decoded pictures* for reference, output reordering, or output delay specified for the *hypothetical reference decoder* in Annex 0.

3.31 **decoder**: An embodiment of a *decoding process*.

3.32 **decoder under test (DUT)**: A *decoder* that is tested for conformance to this Recommendation | International Standard by operating the *hypothetical stream scheduler* to deliver a conforming *bitstream* to the *decoder* and to the *hypothetical reference decoder* and comparing the values and timing of the output of the two *decoders*.

3.33 **decoding order**: The order in which *syntax elements* are processed by the *decoding process*.

3.34 **decoding process**: The process specified in this Recommendation | International Standard that reads a *bitstream* and derives *decoded pictures* from it.

3.35 **display process**: A process not specified in this Recommendation | International Standard having, as its input, the cropped decoded *pictures* that are the output of the *decoding process*.

3.36 **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*.

3.37 **encoder**: An embodiment of an *encoding process*.

3.38 **encoding process**: A process, not specified in this Recommendation | International Standard, that produces a *bitstream* conforming to this Recommendation | International Standard.

3.39 **flag**: A variable that can take one of the two possible values 0 and 1.

3.40 **frequency index**: A one-dimensional or two-dimensional index associated with a *transform coefficient* prior to an *inverse transform* part of the *decoding process*.

3.41 **hypothetical reference decoder (HRD)**: A hypothetical *decoder* model that specifies constraints on the variability of conforming *NAL unit streams* or conforming *byte streams* that an encoding process may produce.

3.42 **hypothetical stream scheduler (HSS)**: A hypothetical delivery mechanism for the timing and data flow of the input of a *bitstream* into the *hypothetical reference decoder*. The HSS is used for checking the conformance of a *bitstream* or a *decoder*.

3.43 **I slice**: A *slice* that is decoded using *intra prediction* only.

3.44 **informative**: A term used to refer to content provided in this Recommendation | International Standard that is not an integral part of this Recommendation | International Standard. Informative content does not establish any mandatory requirements for conformance to this Recommendation | International Standard.

3.45 **instantaneous decoding refresh (IDR) access unit**: An *access unit* in which the *primary coded picture* is an *IDR picture*.

3.46 **instantaneous decoding refresh (IDR) picture**: A *coded picture* for which the variable IdrPicFlag is equal to 1. An IDR picture causes the *decoding process* to mark all *reference pictures* as "unused for reference"

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000283

immediately after the decoding of the IDR picture. All *coded pictures* that follow an IDR picture in *decoding order* can be decoded without *inter prediction* from any *picture* that precedes the IDR picture in *decoding order*. The first *picture* of each *coded video sequence* in *decoding order* is an IDR picture.

3.47 **inter coding**: Coding of a *block*, *macroblock*, *slice*, or *picture* that uses *inter prediction*.

3.48 **inter prediction**: A *prediction* derived from decoded samples of *reference pictures* other than the current *decoded picture*.

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 *slice*.

3.51 **intra slice**: See *I slice*.

3.52 **inverse transform**: A part of the *decoding process* by which a set of *transform coefficients* are converted into spatial-domain values, or by which a set of *transform coefficients* are converted into *DC transform 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 *coded video sequence*, *picture*, *slice*, and *treeblock* layers.

3.54 **leaf**: A terminating node of a tree that is a root node of a tree of depth 0.

3.55 **level**: A defined set of constraints on the values that may be taken by the *syntax elements* and variables of this Recommendation | International Standard. The same set of levels is defined for all *profiles*, with most aspects of the definition of each level being in common across different *profiles*. Individual implementations may, within specified constraints, support a different level for each supported *profile*. In a different context, level is the value of a *transform coefficient* prior to *scaling*.

3.56 **list 0 (list 1) motion vector**: A *motion vector* associated with a *reference index* pointing into *reference picture list 0* (*list 1*).

3.57 **list 0 (list 1, list combination) prediction**: *Inter prediction* of the content of a *slice* using a *reference index* pointing into *reference picture list 0* (*list 1, combination*).

3.58 **luma**: An adjective specifying that a sample array or single sample is representing the monochrome signal related to the primary colours. The symbol or subscript used for luma is Y or L.

NOTE – 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. The symbol L is sometimes used instead of the symbol Y to avoid confusion with the symbol y as used for vertical location.

3.59 **may**: A term used to refer to behaviour that is allowed, but not necessarily required. In some places where the optional nature of the described behaviour is intended to be emphasized, the phrase "may or may not" is used to provide emphasis.

3.60 **memory management control operation**: Seven operations that control *reference picture marking*.

3.61 **motion vector**: A two-dimensional vector used for *inter prediction* that provides an offset from the coordinates in the *decoded picture* to the coordinates in a *reference picture*.

3.62 **must**: A term used in expressing an observation about a requirement or an implication of a requirement that is specified elsewhere in this Recommendation | International Standard. This term is used exclusively in an *informative* context.

3.63 **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*.

3.64 **NAL unit stream**: A sequence of *NAL units*.

3.65 **non-reference picture**: A *picture* coded with nal_ref_idc equal to 0. A *non-reference picture* is not used for *inter prediction* of any other *pictures*.

3.66 **note**: A term used to prefix *informative* remarks. This term is used exclusively in an *informative* context.

3.67 **output order**: The order in which the *decoded pictures* are output from the *decoded picture buffer*.

3.68 **P slice**: A *slice* that may be decoded using *intra prediction* or *inter prediction* using at most one *motion vector* and *reference index* to *predict* the sample values of each *block*.

3.69 **parameter**: A *syntax element* of a *sequence parameter set* or a *picture parameter set*. Parameter is also used as part of the defined term *quantisation parameter*.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000284

3.70 **partitioning**: The division of a set into subsets such that each element of the set is in exactly one of the subsets.

3.71 **picture**: [Ed. (TW) define]

3.72 **picture parameter set**: A *syntax structure* containing *syntax elements* that apply to zero or more entire *coded pictures* as determined by the pic_parameter_set_id *syntax element* found in each *slice header.*

3.73 **picture order count**: [Ed. (TW) re-check] A variable that is associated with each *coded picture* and has a value that is non-decreasing with increasing *picture* position in *output order* relative to the first output *picture* of the previous *IDR picture* in *decoding order* or relative to the first output *picture* of the previous *picture*, in *decoding order*, that contains a *memory management control operation* that marks all *reference pictures* as "unused for reference".

3.74 **prediction**: An embodiment of the *prediction process*.

3.75 **prediction process**: The use of a *predictor* to provide an estimate of the sample value or data element currently being decoded.

3.76 **predictive slice**: See *P slice*.

3.77 **predictor**: A combination of specified values or previously decoded sample values or data elements used in the *decoding process* of subsequent sample values or data elements.

3.78 **primary coded picture**: The coded representation of a *picture* to be used by the *decoding process* for a bitstream conforming to this Recommendation | International Standard. The primary coded picture contains all *treeblocks* of the *picture.* The only *pictures* that have a normative effect on the *decoding process* are primary coded pictures.

3.79 **profile**: A specified subset of the syntax of this Recommendation | International Standard.

3.80 **quadtree**: A *tree* in which a parent node can be split into four child nodes. A child node may become parent node for another plit into four child nodes.

3.81 **quantisation parameter**: A variable used by the *decoding process* for *scaling* of *transform coefficient levels*.

3.82 **random access**: The act of starting the decoding process for a *bitstream* at a point other than the beginning of the stream.

3.83 **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.

3.84 **raw byte sequence payload (RBSP):** A *syntax structure* containing an integer number of *bytes* that is encapsulated in a *NAL unit*. An RBSP is either empty 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.

3.85 **raw byte sequence payload (RBSP) stop bit:** A bit equal to 1 present within 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.*

3.86 **recovery point**: A point in the *bitstream* at which the recovery of an exact or an approximate representation of the *decoded pictures* represented by the *bitstream* is achieved after a *random access* or *broken link*.

3.87 **reference index**: An index into a *reference picture list.*

3.88 **reference picture**: A *picture* with nal_ref_idc not equal to 0. A *reference picture* contains samples that may be used for *inter prediction* in the *decoding process* of subsequent *pictures* in *decoding order.*

3.89 **reference picture list**: A list of *reference pictures* that is used for *uni-prediction* of a *P* or *B slice.* For the *decoding process* of a *P slice,* there is one reference picture list. For the *decoding process* of a *B slice*, there are two reference picture lists (list 0 and list 1) and a reference picture lists combination. [Ed. (TW) the latter may not be true anymore]

3.90 **reference picture list 0**: A *reference picture list* used for *inter prediction* of a *P* or *B slice.* All *inter prediction* used for *P slices* uses reference picture list 0. Reference picture list 0 is one of two *reference picture lists* used for *bi-prediction* for a *B slice*, with the other being *reference picture list.* [Ed. (TW) the latter may not be true anymore]

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000285

3.91 **reference picture list 1**: A *reference picture list* used for *bi-prediction* of a *B slice*. Reference picture list 1 is one of two *reference picture lists* used for *bi-prediction* for a *B slice*, with the other being *reference picture list 0*.

3.92 **reference picture lists combination**: A reference picture list used for uni-prediction of a B slice. Reference picture lists combination is derived from the entries of the reference picture lists 0 and reference picture list 1.

3.93 **reference picture marking**: Specifies, in the bitstream, how the *decoded pictures* are marked for *inter prediction*.

3.94 **reserved**: The term reserved, when used in the clauses specifying some values of a particular *syntax element*, are for future use by ITU-T | ISO/IEC. These values shall not be used in *bitstreams* conforming to this Recommendation | International Standard, but may be used in future extensions of this Recommendation | International Standard by ITU-T | ISO/IEC.

3.95 **residual**: The decoded difference between a *prediction* of a sample or data element and its decoded value.

3.96 **sample aspect ratio**: Specifies, for assisting the *display process*, which is not specified in this Recommendation | International Standard, the ratio between the intended horizontal distance between the columns and the intended vertical distance between the rows of the *luma* sample array in a *picture*. Sample aspect ratio is expressed as *h*:*v*, where *h* is horizontal width and *v* is vertical height (in arbitrary units of spatial distance).

3.97 **scaling**: The process of multiplying *transform coefficient levels* by a factor, resulting in *transform coefficients*.

3.98 **sequence parameter set**: A *syntax structure* containing *syntax elements* that apply to zero or more entire *coded video sequences* as determined by the content of a seq_parameter_set_id *syntax element* found in the *picture parameter set* referred to by the pic_parameter_set_id *syntax element* found in each *slice header*.

3.99 **shall**: A term used to express mandatory requirements for conformance to this Recommendation | International Standard. When used to express a mandatory constraint on the values of *syntax elements* or on the results obtained by operation of the specified *decoding process*, it is the responsibility of the *encoder* to ensure that the constraint is fulfilled. When used in reference to operations performed by the *decoding process*, any *decoding process* that produces identical results to the *decoding process* described herein conforms to the *decoding process* requirements of this Recommendation | International Standard.

3.100 **should**: A term used to refer to behaviour of an implementation that is encouraged to be followed under anticipated ordinary circumstances, but is not a mandatory requirement for conformance to this Recommendation | International Standard.

3.101 **slice**: An integer number of *treeblocks* ordered consecutively in the *raster scan*. For the *primary coded picture*, the division of each *picture* into slices is a *partitioning*. The *treeblock addresses* are derived from the first *treeblock address* in a slice (as represented in the *slice header*).

3.102 **slice header**: A part of a coded *slice* containing the data elements pertaining to the first or all *macroblocks* represented in the *slice*.

3.103 **source**: Term used to describe the video material or some of its attributes before encoding.

3.104 **start code prefix**: A unique sequence of three *bytes* equal to 0x000001 embedded in the *byte stream* 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 prevented within *NAL units* by the inclusion of *emulation prevention bytes*.

3.105 **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*. Within an SODB, the left-most bit is considered to be the first and most significant bit, and the right-most bit is considered to be the last and least significant bit.

3.106 **syntax element**: An element of data represented in the *bitstream*.

3.107 **syntax structure**: Zero or more *syntax elements* present together in the *bitstream* in a specified order.

3.108 **transform coefficient**: A scalar quantity, considered to be in a frequency domain, that is associated with a particular one-dimensional or two-dimensional *frequency index* in an *inverse transform* part of the *decoding process*.

3.109 **transform coefficient level**: An integer quantity representing the value associated with a particular two-dimensional frequency index in the *decoding process* prior to *scaling* for computation of a *transform coefficient* value.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000286

3.110 **tree**: A tree is a finite ste of nodes with a unique root node. A terminating node is called a *leaf*.

3.111 **treeblock**: A NxN *block* of *luma* samples and two corresponding *blocks* of *chroma* samples of a *picture* that has three sample arrays, or a NxN *block* of samples of a monochrome *picture* or a *picture* that is coded using three separate colour planes. The division of a *slice* into treeblocks is a *partitioning*.

3.112 **treeblock address**: A treeblock address is the index of a *treeblock* in a *treeblock raster scan* of the *picture* starting with zero for the top-left *treeblock* in a *picture*.

3.113 **treeblock location**: The two-dimensional coordinates of a *treeblock* in a *picture* denoted by (x, y). For the top left *treeblock* of the *picture* (x, y) is equal to (0, 0). x is incremented by 1 for each *treeblock* column from left to right. The value of y is incremented by 1 for each *treeblock* row from top to bottom.

3.114 **treeblock partition**: A *block* of *luma* samples and two corresponding *blocks* of *chroma* samples resulting from a *partitioning* of a *treeblock* for *inter prediction* for a *picture* that has three sample arrays or a *block* of *luma* samples resulting from a *partitioning* of a *treeblock* for *inter prediction* for a monochrome *picture* or a *picture* that is coded using three separate colour planes.

3.115 **universal unique identifier (UUID)**: An identifier that is unique with respect to the space of all universal unique identifiers.

3.116 **unspecified:** The term unspecified, when used in the clauses specifying some values of a particular *syntax element*, indicates that the values have no specified meaning in this Recommendation | International Standard and will not have a specified meaning in the future as an integral part of this Recommendation | International Standard.

3.117 **variable length coding (VLC)**: A reversible procedure for entropy coding that assigns shorter bit strings to *symbols* expected to be more frequent and longer bit strings to *symbols* expected to be less frequent.

3.118 **VCL NAL unit**: A collective term for *coded slice NAL units*.

3.119 **zig-zag scan**: A specific sequential ordering of *transform coefficient levels* from (approximately) the lowest spatial frequency to the highest.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-based Adaptive Variable Length Coding
CBR	Constant Bit Rate
CDR	Clean Decoding Refresh
CPB	Coded Picture Buffer
DPB	Decoded Picture Buffer
DUT	Decoder under test
FIFO	First-In, First-Out
HRD	Hypothetical Reference Decoder
HSS	Hypothetical Stream Scheduler
IDR	Instantaneous Decoding Refresh
LSB	Least Significant Bit
MSB	Most Significant Bit
NAL	Network Abstraction Layer
RBSP	Raw Byte Sequence Payload
SEI	Supplemental Enhancement Information
SODB	String Of Data Bits
TB	Treeblock

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000287

UUID	Universal Unique Identifier
VBR	Variable Bit Rate
VCL	Video Coding Layer
VLC	Variable Length Coding
VUI	Video Usability Information

5 Conventions

NOTE – The mathematical operators used in this Specification are similar to those used in the C programming language. However, integer division and arithmetic shift operations 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 prefix operator)
*	Multiplication, including matrix multiplication
x^y	Exponentiation. Specifies x to the power of y. In other contexts, such notation is used for superscripting not intended for interpretation 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.
$\frac{x}{y}$	Used to denote division in mathematical equations where no truncation or rounding is intended.
$\sum_{i=x}^{y} f(i)$	The summation of f(i) with i taking all integer values from x up to and including y.
x % y	Modulus. Remainder of x divided by y, defined only for integers x and y with x >= 0 and y > 0.

5.2 Logical operators

The following logical operators are defined as follows:

x && y	Boolean logical "and" of x and y.
x || y	Boolean logical "or" of x and y.
!	Boolean logical "not".
x ? y : z	If x is TRUE or not equal to 0, evaluates to the value of y; otherwise, evaluates to the value of z.

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.

When a relational operator is applied to a syntax element or variable that has been assigned the value "na" (not applicable), the value "na" is treated as a distinct value for the syntax element or variable. The value "na" is considered not to be equal to any other value.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000288

5.4 Bit-wise operators

The following bit-wise operators are defined as follows:

- & Bit-wise "and". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.
- | Bit-wise "or". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.
- ^ Bit-wise "exclusive or". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.
- x >> y Arithmetic right shift of a two's complement integer representation of x by y binary digits. This function is defined only for positive integer values of y. Bits shifted into the MSBs as a result of the right shift have a value equal to the MSB of x prior to the shift operation.
- x << y Arithmetic left shift of a two's complement integer representation of x by y binary digits. This function is defined only for positive integer values of y. Bits shifted into the LSBs as a result of the left shift have a value equal to 0.

5.5 Assignment operators

The following arithmetic operators are defined as follows:

- = Assignment operator.
- ++ 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 value 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 Range notation

The following notation is used to specify a range of values:

- x = y..z x takes on integer values starting from y to z, inclusive, with x, y, and z being integer numbers.

5.7 Mathematical functions

The following mathematical functions are defined as follows:

$$Abs(x) = \begin{cases} x & ; \quad x >= 0 \\ -x & ; \quad x < 0 \end{cases} \tag{5-1}$$

Ceil(x) the smallest integer greater than or equal to x. (5-2)

$$Clip1_Y(x) = Clip3(0, (1 << BitDepth_Y) - 1, x) \tag{5-3}$$

$$Clip1_C(x) = Clip3(0, (1 << BitDepth_C) - 1, x) \tag{5-4}$$

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000289

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases} \tag{5-5}$$

Floor(x) the greatest integer less than or equal to x. (5-6)

$$\text{InverseRasterScan}(a, b, c, d, e) = \begin{cases} (a\,\%\,(d/b))*b & ; \quad e == 0 \\ (a/(d/b))*c & ; \quad e == 1 \end{cases} \tag{5-7}$$

Log2(x) returns the base-2 logarithm of x. (5-8)

Log10(x) returns the base-10 logarithm of x. (5-9)

Median(x, y, z) = x + y + z − Min(x, Min(y, z)) − Max(x, Max(y, z)) (5-10)

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x <= y \\ y & ; \quad x > y \end{cases} \tag{5-11}$$

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x >= y \\ y & ; \quad x < y \end{cases} \tag{5-12}$$

Round(x) = Sign(x) * Floor(Abs(x) + 0.5) (5-13)

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x >= 0 \\ -1 & ; \quad x < 0 \end{cases} \tag{5-14}$$

$$\text{Sqrt}(x) = \sqrt{x} \tag{5-15}$$

5.8 Order of operation precedence

When order of precedence in an expression is not indicated explicitly by use of parentheses, the following rules apply:

- operations of a higher precedence are evaluated before any operation of a lower precedence,
- operations of the same precedence are evaluated sequentially from left to right.

Table 5-1~~Table 5-1~~ specifies the precedence of operations from highest to lowest; a higher position in the table indicates a higher precedence.

NOTE – For those operators that are also used in the C programming language, the order of precedence used in this Specification is the same as used in the C programming language.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000290

Table 5-1 – Operation precedence from highest (at top of table) to lowest (at bottom of table)

operations (with operands x, y, and z)
"x++", "x– –"
"!x", "–x" (as a unary prefix operator)
x^y
"x * y", "x / y", "x ÷ y", "$\frac{x}{y}$", "x % y"
"x + y", "x – y" (as a two-argument operator), "$\sum_{i=x}^{y} f(i)$"
"x << y", "x >> y"
"x < y", "x <= y", "x > y", "x >= y"
"x = = y", "x != y"
"x & y"
"x | y"
"x && y"
"x | | y"
"x ? y : z"
"x = y", "x += y", "x –= y"

5.9 Variables, syntax elements, and tables

Syntax elements in the bitstream are represented in **bold** type. Each syntax element is described by its name (all lower case letters with underscore characters), its one or two syntax categories, and one or two descriptors for its method of coded representation. The decoding process behaves according to the value of the syntax element and to the values of previously decoded syntax elements. When a value of a syntax element is used in the syntax tables or the text, it appears in regular (i.e., not bold) type.

In some cases the syntax tables may use the values of other variables derived from syntax elements values. Such variables appear in the syntax tables, or text, named by a mixture of lower case and upper case letter and without any underscore characters. Variables starting with an upper case letter are derived for the decoding of the current syntax structure and all depending syntax structures. Variables starting with an upper case letter may be used in the decoding process for later syntax structures without mentioning the originating syntax structure of the variable. Variables starting with a lower case letter are only used within the subclause in which they are derived.

In some cases, "mnemonic" names for syntax element values or variable values are used interchangeably with their numerical values. Sometimes "mnemonic" names are used without any associated numerical values. The association of values and names is specified in the text. The names are constructed from one or more groups of letters separated by an underscore character. Each group starts with an upper case letter and may contain more upper case letters.

NOTE – The syntax is described in a manner that closely follows the C-language syntactic constructs.

Functions that specify properties of the current position in the bitstream are referred to as syntax functions. These functions are specified in subclause 0 and assume the existence of a bitstream pointer with an indication of the position of the next bit to be read by the decoding process from the bitstream. Syntax functions are described by their names, which are constructed as syntax element names and end with left and right round parentheses including zero or more variable names (for definition) or values (for usage), separated by commas (if more than one variable).

Functions that are not syntax functions (including mathematical functions specified in subclause 5.7) are described by their names, which start with an upper case letter, contain a mixture of lower and upper case letters without any underscore character, and end with left and right parentheses including zero or more variable names (for definition) or values (for usage) separated by commas (if more than one variable).

A one-dimensional array is referred to as a list. A two-dimensional array is referred to as a matrix. Arrays can either be syntax elements or variables. Subscripts or square parentheses are used for the indexing of arrays. In reference to a visual depiction of a matrix, the first subscript is used as a row (vertical) index and the second subscript is used as a column

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000291

(horizontal) index. The indexing order is reversed when using square parentheses rather than subscripts for indexing. Thus, an element of a matrix s at horizontal position x and vertical position y may be denoted either as s[x, y] or as s_{yx}.

Binary notation is indicated by enclosing the string of bit values by single quote marks. For example, '01000001' represents an eight-bit string having only its second and its last bits (counted from the most to the least significant bit) equal to 1.

Hexadecimal notation, indicated by prefixing the hexadecimal number by "0x", may be used instead of binary notation when the number of bits is an integer multiple of 4. For example, 0x41 represents an eight-bit string having only its second and its last bits (counted from the most to the least significant bit) equal to 1.

Numerical values not enclosed in single quotes and not prefixed by "0x" are decimal values.

A value equal to 0 represents a FALSE condition in a test statement. The value TRUE is represented by any value different from zero.

5.10 Text description of logical operations

In the text, a statement of logical operations as would be described in pseudo-code as

```
if( condition 0 )
  statement 0
else if ( condition 1 )
  statement 1
…
else /* informative remark on remaining condition */
  statement n
```

may be described in the following manner:

... as follows / ... the following applies.

- If condition 0, statement 0
- Otherwise, if condition 1, statement 1
- …
- Otherwise (informative remark on remaining condition), statement n

Each "If ... Otherwise, if ... Otherwise, ..." statement in the text is introduced with "... as follows" or "... the following applies" immediately followed by "If ... ". The last condition of the "If ... Otherwise, if ... Otherwise, ..." is always an "Otherwise, ...". Interleaved "If ... Otherwise, if ... Otherwise, ..." statements can be identified by matching "... as follows" or "... the following applies" with the ending "Otherwise, ...".

In the text, a statement of logical operations as would be described in pseudo-code as

```
if( condition 0a  &&  condition 0b )
  statement 0
else if ( condition 1a  ||  condition 1b )
  statement 1
…
else
  statement n
```

may be described in the following manner:

... as follows / ... the following applies.

- If all of the following conditions are true, statement 0
  - condition 0a
  - condition 0b
- Otherwise, if any of the following conditions are true, statement 1
  - condition 1a
  - condition 1b
- …
- Otherwise, statement n

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000292

In the text, a statement of logical operations as would be described in pseudo-code as:

```
if( condition 0 )
  statement 0
if ( condition 1 )
  statement 1
```

may be described in the following manner:

When condition 0, statement 0

When condition 1, statement 1

5.11 Processes

Processes are used to describe the decoding of syntax elements. A process has a separate specification and invoking. All syntax elements and upper case variables that pertain to the current syntax structure and depending syntax structures are available in the process specification and invoking. A process specification may also have a lower case variable explicitly specified as the input. Each process specification has explicitly specified an output. The output is a variable that can either be an upper case variable or a lower case variable.

When invoking a process, the assignment of variables is specified as follows.

- If the variables at the invoking and the process specification do not have the same name, the variables are explicitly assigned to lower case input or output variables of the process specification.
- Otherwise (the variables at the invoking and the process specification have the same name), assignment is implied.

In the specification of a process, a specific macroblock may be referred to by the variable name having a value equal to the address of the specific macroblock.

6 Source, coded, decoded and output data formats, scanning processes, and neighbouring relationships

6.1 Bitstream formats

This subclause specifies the relationship between the NAL unit stream and byte stream, either 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 structures 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 ordering 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 of 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 pictures that is given via the bitstream.

The video source that is represented by the bitstream is a sequence of pictures in decoding order.

The source and decoded pictures are each comprised of one or more sample arrays:

- Luma (Y) only (monochrome).
- Luma and two chroma (YCbCr or YCgCo).
- Green, Blue and Red (GBR, also known as RGB).
- Arrays representing other unspecified monochrome or tri-stimulus colour samplings (for example, YZX, also known as XYZ).

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000293

For convenience of notation and terminology in this Specification, the variables and terms associated with these arrays are referred to as luma (or L or Y) and chroma, where the two chroma arrays are referred to as Cb and Cr; regardless of the actual colour representation method in use. The actual colour representation method in use can be indicated in syntax that is specified in Annex E.

The variables SubWidthC, and SubHeightC are specified in Table 6-1~~Table 6-1~~, depending on the chroma format sampling structure, which is specified through chroma_format_idc and separate_colour_plane_flag. An entry marked as "-" in Table 6-1~~Table 6-1~~ denotes an undefined value for SubWidthC or SubHeightC. Other values of chroma_format_idc, SubWidthC, and SubHeightC may be specified in the future by ITU-T | ISO/IEC.

Table 6-~~16-1~~1 – SubWidthC, and SubHeightC values derived from chroma_format_idc and separate_colour_plane_flag

chroma_format_idc	separate_colour_plane_flag	Chroma Format	SubWidthC	SubHeightC
0	0	monochrome	-	-
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	-	-

In monochrome sampling there is only one sample array, which is nominally considered the luma array.

In 4:2:0 sampling, each of the two chroma arrays has half the height and half the width of the luma array.

In 4:2:2 sampling, each of the two chroma arrays has the same height and half the width of the luma array.

In 4:4:4 sampling, depending on the value of separate_colour_plane_flag, the following applies.

– If separate_colour_plane_flag is equal to 0, each of the two chroma arrays has the same height and width as the luma array.

– Otherwise (separate_colour_plane_flag is equal to 1), the three colour planes are separately processed as monochrome sampled pictures.

The number of bits necessary for the representation of each of the samples in the luma and chroma arrays in a video sequence is in the range of 8 to 14, and the number of bits used in the luma array may differ from the number of bits used in the chroma arrays.

When the value of chroma_format_idc is equal to 1, the nominal vertical and horizontal relative locations of luma and chroma samples in pictures are shown in Figure 6-1~~Figure 6-1~~. Alternative chroma sample relative locations may be indicated in video usability information (see Annex E).

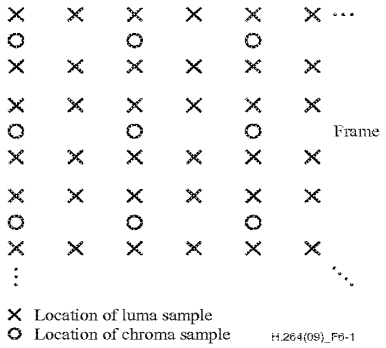


Figure 6-1 – Nominal vertical and horizontal locations of 4:2:0 luma and chroma samples in a picture [Ed. Re-draw figure]

When the value of chroma_format_idc is equal to 2, the chroma samples are co-sited with the corresponding luma samples and the nominal locations in a picture are as shown in Figure 6-2~~6-2~~.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000294

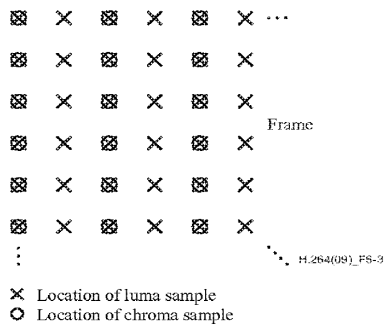


Figure 6-2 – Nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a picture [Ed. Re-draw figure]

When the value of chroma_format_idc is equal to 3, all array samples are co-sited for all cases of pictures and the nominal locations in a picture are as shown in Figure 6-36-3.

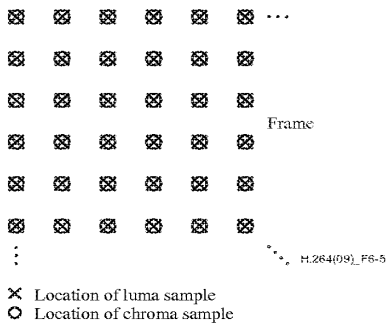


Figure 6-3 – Nominal vertical and horizontal locations of 4:4:4 luma and chroma samples in a picture [Ed. Re-draw figure]

The samples are processed in units of treeblocks. The luma array for each treeblock in samples in both width and height is

$$TbSize = 64 \quad (6\text{-}1)$$
$$Log2TbSize = 6 \quad (6\text{-}2)$$

The variables TbWidthC and TbHeightC, which specify the width and height, respectively, of the chroma arrays for each treeblock, are derived as follows.

– If chroma_format_idc is equal to 0 (monochrome) or separate_colour_plane_flag is equal to 1, TbWidthC and TbHeightC are both equal to 0.

– Otherwise, TbWidthC and TbHeightC are derived as

$$TbWidthC = TbSize / SubWidthC \quad (6\text{-}3)$$
$$TbHeightC = TbSize / SubHeightC \quad (6\text{-}4)$$

6.3 Spatial subdivision of pictures and slices

This subclause specifies how a picture is partitioned into slices and treeblocks. Pictures are divided into slices. A slice is a sequence of treeblocks.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000295

Each treeblock is comprised of one Ntb x Ntb luma array and, when the chroma sampling format is not equal to 4:0:0 and separate_colour_plane_flag is equal to 0, two corresponding chroma sample arrays. When separate_colour_plane_flag is equal to 1, each treeblock is comprised of one Ntb x Ntb luma or chroma sample array. For example, a picture may be divided into two slices as shown in Figure 6-4~~Figure 6-4~~.

When a picture is coded using three separate colour planes (separate_colour_plane_flag is equal to 1), a slice contains only treeblocks of one colour component being identified by the corresponding value of colour_plane_id, and each colour component array of a picture consists of slices having the same colour_plane_id value. Coded slices with different values of colour_plane_id within an access unit can be interleaved with each other under the constraint that for each value of colour_plane_id, the coded slice NAL units with that value colour_plane_id shall be in the order of increasing treeblock address for the first treeblock of each coded slice NAL unit.

NOTE – When separate_colour_plane_flag is equal to 0, each treeblock of a picture is contained in exactly one slice. When separate_colour_plane_flag is equal to 1, each treeblock of a colour component is contained in exactly one slice (i.e., information for each treeblock of a picture is present in exactly three slices and these three slices have different values of colour_plane_id).

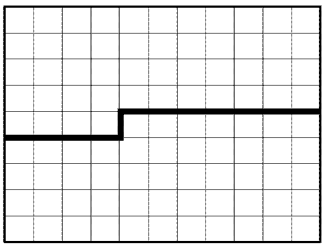


Figure 6-4 – A picture with 11 by 9 treeblocks that is partitioned into two slices

Each treeblock is assigned a partition signalling to identify the block sizes for intra or inter prediction and for transform coding. The partitioning is a recursive quadtree partitioning. The root of the quadtree is associated with the treeblock. The quadtree is split until a leaf is reached, which is referred to as the coding node. The coding node is the root node of two trees, the prediction tree and the transform tree.

~~[Ed. (JW) What is a "unit"?]~~

The prediction tree specifies the position and size of prediction blocks. The prediction tree and associated prediction data are referred to as prediction unit.

The transform tree specifies the position and size of transform blocks. The transform tree and associated transform data are referred to as transform unit.

The splitting information for luma and chroma is identical for the prediction tree and may or may not be identical for the transform tree.

The coding node and the associated prediction and transform units form together a coding unit.

[Ed.: (WJ) block = a rectangular 2D array (one component), unit = collective term for specifying information for both luma and chroma. Don't use ther term 'unit' by itself – always use the term 'unit' with prefix – coding unit, prediction unit or transform unit.]

6.4 Inverse scanning processes and derivation processes for neighbours

This subclause specifies inverse scanning processes; i.e., the mapping of indices to locations, and derivation processes for neighbours.

6.4.1 Inverse treeblock scanning process

Input to this process is a treeblock address tbAddr.

Output of this process is the location (x, y) of the upper-left luma sample for the treeblock with address tbAddr relative to the upper-left sample of the picture.

The inverse treeblock scanning process is specified as follows.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000296

x = InverseRasterScan(tbAddr, 16, 16, $PicWidthInSamples_L$, 0) (6-5)

y = InverseRasterScan(tbAddr, 16, 16, $PicWidthInSamples_L$, 1) (6-6)

[Ed. (TW): The text found in JCVTVC-B205 that relates to the rest of this clause requires substantial editorial and formatting work. It cannot be imported as is.]

6.4.2 Derivation process for the availability of treeblock addresses

Input to this process is a treeblock address tbAddr.

Output of this process is the availability of the treeblock tbAddr.

NOTE – The meaning of availability is determined when this process is invoked.

The treeblock is marked as available, unless one of the following conditions is true in which case the treeblock shall be marked as not available:

- tbAddr < 0
- tbAddr > CurrTbAddr
- the treeblock with address tbAddr belongs to a different slice than the treeblock with address CurrTbAddr and the lightweight_slice_flag of the slice containing the treeblock with address CurrTbAddr is equal to 0.

[Ed. (WJ): do we need to separate the use of slice and lightweight slice? E.g. Is lightweight slice a kind of slice or sub-slice? We may need the concrete definitions]

6.5 Zig-zag scanning array initialisation process

Input to this process is a block size blkSize.

Output of this process is the array ZigZag[pos][comp]. The array index pos specify the scan position ranging from 0 to (blkSize * blkSize) – 1. The array index comp equal to 0 specifies the horizontal component and the array index comp equal to 1 specifies the vertical component. The array ZagZag is derived as follows.

```
ZigZag[ 0 ][ 0 ] = 0
ZigZag[ 0 ][ 1 ] = 0
i = 1
x = 1
y = 0
stopLoop = ( blkSize == 1 ) ? true : false
while( !stopLoop ) {
    while( x >= 0 ) {
        if (x >= 0 && x < blkSize && y >= 0 && y < blkSize ) {
            ZigZag[ i ][ 0 ] = x
            ZigZag[ i ][ 1 ] = y
            i++
        }
        x--
        y++
    }
    x = 0
    while( y >= 0 ) {
        if (x >= 0 && x < blkSize && y >= 0 && y < blkSize ) {
            ZigZag[ i ][ 0 ] = x
            ZigZag[ i ][ 1 ] = y
            i++
        }
        x++
        y--
    }
    y = 0
    if ( i >= blkSize * blkSize )
        stopLoop = true
}
```


Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000297

6.6 Scanning order array initialisation process

Input to this process is a block size blkSize.

Output of this process is the array ScanOrder[scanIdx][pos][comp]. The array index scanIdx equal to 0 specifies a zig-zag scan as specified in subclause 오류! 참조 원본을 찾을 수 없습니다.6.5, scanIdx equal to 1 specifies a horizontal scan and scanIdx equal to 2 specifies a vertical scan. The array index pos specifies the scan position ranging from 0 to (blkSize * blkSize) − 1. The array index comp equal to 0 specifies the horizontal component and the array index comp equal to 1 specifies the vertical component. The array ScanOrder is derived as follows.

```
ScanOrder[0] = ZigZag

i = 0
y = 0
while( y < blkSize ) {
    x = 0
    while( x < blkSize ) {
        ScanOrder[ 1 ][ i ][ 0 ] = x
        ScanOrder[ 1 ][ i ][ 1 ] = y
        x++
        i++
    }
    y++
}

i = 0
x = 0
while( x < blkSize ) {
    y = 0
    while( y < blkSize ) {
        ScanOrder[ 2 ][ i ][ 0 ] = x
        ScanOrder[ 2 ][ i ][ 1 ] = y
        y++
        i++
    }
    x++
}
```

7 Syntax and semantics

7.1 Method of specifying syntax in tabular form

The syntax tables specify a superset of the syntax of all allowed bitstreams. Additional constraints on the syntax may be specified, either directly or indirectly, in other clauses.

NOTE – An actual decoder should implement means for identifying entry points into the bitstream and means to identify and handle non-conforming bitstreams. The methods for identifying and handling errors and other such situations are not specified here.

The following table lists examples of pseudo code used to describe the syntax. When **syntax_element** appears, it specifies that a syntax element is parsed from the bitstream and the bitstream pointer is advanced to the next position beyond the syntax element in the bitstream parsing process.

	C	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		

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000298

statement		
…		
}		
/* A "while" structure specifies a test of whether a condition is true, and if true, specifies evaluation of a statement (or compound statement) repeatedly until the condition is no longer true */		
while(condition)		
statement		
/* A "do … while" structure specifies evaluation of a statement once, followed by a test of whether a condition is true, and if true, specifies repeated evaluation of the statement until the condition is no longer true */		
do		
statement		
while(condition)		
/* An "if … else" structure specifies a test of whether a condition is true, and if the condition is true, specifies evaluation of a primary statement, otherwise, specifies 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 specifies evaluation of an initial statement, followed by a test of a condition, and if the condition is true, specifies 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 Specification of syntax 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 decoding process from the bitstream.

byte_aligned( ) is specified as follows.

- 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, the return value of byte_aligned( ) is equal to TRUE.
- Otherwise, the return value of byte_aligned( ) is equal to FALSE.

more_data_in_byte_stream( ), which is used only in the byte stream NAL unit syntax structure specified in Annex B, is specified as follows.

- If more data follow in the byte stream, the return value of more_data_in_byte_stream( ) is equal to TRUE.
- Otherwise, the return value of more_data_in_byte_stream( ) is equal to FALSE.

more_rbsp_data( ) is specified as follows.

- If there is no more data in the RBSP, the return value of more_rbsp_data( ) is equal to FALSE.
- Otherwise, the RBSP data is searched for the last (least significant, right-most) bit equal to 1 that is present in the RBSP. Given the position of this bit, which is the first bit (rbsp_stop_one_bit) of the rbsp_trailing_bits( ) syntax structure, the following applies.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000299

- If there is more data in an RBSP before the rbsp_trailing_bits( ) syntax structure, the return value of more_rbsp_data( ) is equal to TRUE.
- Otherwise, the return value of more_rbsp_data( ) is equal to FALSE.

The method for enabling determination of whether there is more data in the RBSP is specified by the application (or in Annex B for applications that use the byte stream format).

more_rbsp_trailing_data( ) is specified as follows.

- If there is more data in an RBSP, the return value of more_rbsp_trailing_data( ) is equal to TRUE.
- Otherwise, the return value of more_rbsp_trailing_data( ) is equal to FALSE.

next_bits(n) provides the next bits in the bitstream for comparison purposes, without advancing the bitstream pointer. Provides a look at the next n bits in the bitstream with n being its argument. When used within the byte stream as specified in Annex B, next_bits(n) returns a value of 0 if fewer than n bits remain within the byte stream.

read_bits(n) reads the next n bits from the bitstream and advances the bitstream pointer by n bit positions. When n is equal to 0, read_bits(n) is specified to return a value equal to 0 and to not advance the bitstream pointer.

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_flag is equal to 0 and the right descriptor applies when entropy_coding_mode_flag 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~~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. When n is "v" in the syntax table, 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 integer representation with most significant bit written first.
- me(v): mapped Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in subclause 9.1~~9.1~~.
- 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~~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~~9.1~~.
- u(n): unsigned integer using n bits. When n is "v" in the syntax table, 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~~9.1~~.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000300

7.3 Syntax in tabular form

7.3.1 NAL unit syntax

nal_unit(NumBytesInNALunit) {	**Descriptor**
forbidden_zero_bit	f(1)
nal_ref_idc	u(2)
nal_unit_type	u(5)
NumBytesInRBSP = 0	
nalUnitHeaderBytes = 1	
if(nal_unit_type == 1 || nal_unit_type == 4 || nal_unit_type == 5) {	
temporal_id	u(3)
output_flag	u(1)
reserved_one_4bits	u(4)
nalUnitHeaderBytes += 1	
}	
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {	
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {	
rbsp_byte[NumBytesInRBSP++ **]**	b(8)
rbsp_byte[NumBytesInRBSP++ **]**	b(8)
i += 2	
emulation_prevention_three_byte /* equal to 0x03 */	f(8)
} else	
rbsp_byte[NumBytesInRBSP++ **]**	b(8)
}	
}	

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000301

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( ) {	Descriptor
profile_idc	u(8)
reserved_zero_8bits /* equal to 0 */	u(8)
level_idc	u(8)
seq_parameter_set_id	ue(v)
max_temporal_layers_minus1	u(3)
pic_width_in_luma_samples	u(16)
pic_height_in_luma_samples	u(16)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
pcm_bit_depth_luma_minus1	u(4)
pcm_bit_depth_chroma_minus1	u(4)
log2_max_frame_num_minus4	ue(v)
pic_order_cnt_type	ue(v)
if(pic_order_cnt_type == 0)	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
else if(pic_order_cnt_type == 1) {	
delta_pic_order_always_zero_flag	u(1)
offset_for_non_ref_pic	se(v)
num_ref_frames_in_pic_order_cnt_cycle	ue(v)
for(i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++)	
offset_for_ref_frame[i]	se(v)
}	
max_num_ref_frames	ue(v)
gaps_in_frame_num_value_allowed_flag	u(1)
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
log2_min_pcm_coding_block_size_minus3	ue(v)
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
chroma_pred_from_luma_enabled_flag	u(1)
loop_filter_across_slice_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
pcm_loop_filter_disable_flag	u(1)
cu_qp_delta_enabled_flag	u(1)
temporal_id_nesting_flag	u(1)
rbsp_trailing_bits( )	
}	

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000302

7.3.2.2 Picture parameter set RBSP syntax

pic_parameter_set_rbsp( ) {	Descriptor
pic_parameter_set_id	ue(v)
seq_parameter_set_id	ue(v)
entropy_coding_mode_flag	u(1)
num_temporal_layer_switching_point_flags	ue(v)
for(i = 0; i < num_temporal_layer_switching_point_flags; i++)	
temporal_layer_switching_point_flag[i]	u(1)
num_ref_idx_l0_default_active_minus1	ue(v)
num_ref_idx_l1_default_active_minus1	ue(v)
pic_init_qp_minus26 /* relative to 26 */	se(v)
constrained_intra_pred_flag	u(1)
slice_granularity	u(2)
shared_pps_info_enabled_flag	u(1)
if(shared_pps_info_enabled_flag)	
if(adaptive_loop_filter_enabled_flag)	
alf_param( )	
if(cu_qp_delta_enabled_flag)	
max_cu_qp_delta_depth	u(4)
rbsp_trailing_bits( )	
}	

7.3.2.3 Supplemental enhancement information RBSP syntax

sei_rbsp( ) {	Descriptor
do	
sei_message( )	
while(more_rbsp_data( ))	
rbsp_trailing_bits( )	
}	

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000303

7.3.2.3.1 Supplemental enhancement information message syntax

sei_message( ) {	Descriptor
payloadType = 0	
while(next_bits(8) = = 0xFF) {	
ff_byte /* equal to 0xFF */	f(8)
payloadType += 255	
}	
last_payload_type_byte	u(8)
payloadType += last_payload_type_byte	
payloadSize = 0	
while(next_bits(8) = = 0xFF) {	
ff_byte /* equal to 0xFF */	f(8)
payloadSize += 255	
}	
last_payload_size_byte	u(8)
payloadSize += last_payload_size_byte	
sei_payload(payloadType, payloadSize)	
}	

7.3.2.4 Access unit delimiter RBSP syntax

access_unit_delimiter_rbsp( ) {	Descriptor
primary_pic_type	u(3)
rbsp_trailing_bits( )	
}	

7.3.2.5 Filler data RBSP syntax

filler_data_rbsp( ) {	Descriptor
while(next_bits(8) = = 0xFF)	
ff_byte /* equal to 0xFF */	f(8)
rbsp_trailing_bits( )	
}	

7.3.2.6 Slice layer RBSP syntax

slice_layer_rbsp( ) {	Descriptor
slice_header( )	
slice_data( )	
rbsp_slice_trailing_bits( )	
}	

서식 있음: 제목 5, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000304

7.3.2.7 RBSP slice trailing bits syntax

rbsp_slice_trailing_bits( ) {	Descriptor
rbsp_trailing_bits( )	
if(entropy_coding_mode_flag)	
while(more_rbsp_trailing_data( ))	
cabac_zero_word /* equal to 0x0000 */	f(16)
}	

7.3.2.8 RBSP trailing bits syntax

rbsp_trailing_bits( ) {	Descriptor
rbsp_stop_one_bit /* equal to 1 */	f(1)
while(!byte_aligned( ))	
rbsp_alignment_zero_bit /* equal to 0 */	f(1)
}	

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000305

7.3.3 Slice header syntax

slice_header( ) {	Descriptor
first_tb_in_slice	ue(v)
entropy_slice_flag**lightweight_slice_flag**	u(1)
if(!entropy**lightweight**_slice_flag) {	
slice_type	ue(v)
pic_parameter_set_id	ue(v)
frame_num	u(v)
if(IdrPicFlag)	
idr_pic_id	ue(v)
if(pic_order_cnt_type == 0)	
pic_order_cnt_lsb /*	u(v)
if(slice_type == P || slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
ref_pic_list_modification( )	
ref_pic_list_combination( )	
if(nal_ref_idc != 0)	
dec_ref_pic_marking( )	
}	
if(entropy_coding_mode_flag && slice_type != I)	
cabac_init_idc	ue(v)
first_slice_in_pic_flag	u(1)
if(first_slice_in_pic_flag == 0)	
slice_address	u(v)
if(!lightweight_slice_flag) {	
slice_qp_delta	se(v)
if(sample_adaptive_offset_enabled_flag)	
sao_param()	
if(deblocking_filter_control_present_flag) {	
disable_deblocking_filter_idc	
if(disable_deblocking_filter_idc != 1) {	
slice_alpha_c0_offset_div2	
slice_beta_offset_div2	
}	

서식 있음: 글꼴: 굵게

서식 있음: 글꼴: 굵게

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000306

}	
if(slice_type = = B)	
collocated_from_l0_flag	u(1)
if(adaptive_loop_filter_enabled_flag) {	
if(!shared_pps_info_enabled_flag)	
alf_param( )	
alf_cu_control_param( )	
}	
} ~~else~~	
~~if(entropy_coding_mode_flag && slice_type != I)~~	
~~**cabac_init_idc**~~	~~ue(v)~~
}	

7.3.3.1 Reference picture list modification syntax

ref_pic_list_modification( ) {	**Descriptor**
if(slice_type % 5 != 2 && slice_type % 5 != 4) {	
ref_pic_list_modification_flag_l0	u(1)
if(ref_pic_list_modification_flag_l0)	
do {	
modification_of_pic_nums_idc	ue(v)
if(modification_of_pic_nums_idc = = 0 || modification_of_pic_nums_idc = = 1)	
abs_diff_pic_num_minus1	ue(v)
else if(modification_of_pic_nums_idc = = 2)	
long_term_pic_num	ue(v)
} while(modification_of_pic_nums_idc != 3)	
}	
if(slice_type % 5 = = 1) {	
ref_pic_list_modification_flag_l1	u(1)
if(ref_pic_list_modification_flag_l1)	
do {	
modification_of_pic_nums_idc	ue(v)
if(modification_of_pic_nums_idc = = 0 || modification_of_pic_nums_idc = = 1)	
abs_diff_pic_num_minus1	ue(v)
else if(modification_of_pic_nums_idc = = 2)	
long_term_pic_num	ue(v)
} while(modification_of_pic_nums_idc != 3)	
}	
}	

7.3.3.2 Reference picture lists combination syntax

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000307

ref_pic_list_combination( ) {	C	Descriptor
if(slice_type % 5 = = 1) { // b slice		
ref_pic_list_combination_flag	2	u(1)
if(ref_pic_list_combination_flag) {		
num_ref_idx lc_active_minus1	2	ue(v)
ref_pic_list_modification_flag_lc	2	u(1)
if(ref_pic_list_modification_flag_lc)		
for (i =0; i <= num_ref_idx_lc_active_minus1; i++) {		
pic_from_list_0_flag	2	u(1)
ref_idx_list_curr	2	ue(v)
}		
}		
}		
}		

7.3.3.3 Decoded reference picture marking syntax

dec_ref_pic_marking( ) {	Descriptor
if(IdrPicFlag) {	
no_output_of_prior_pics_flag	u(1)
long_term_reference_flag	u(1)
} else {	
adaptive_ref_pic_marking_mode_flag	u(1)
if(adaptive_ref_pic_marking_mode_flag)	
do {	
memory_management_control_operation	ue(v)
if(memory_management_control_operation = = 1 || memory_management_control_operation = = 3)	
difference_of_pic_nums_minus1	ue(v)
if(memory_management_control_operation = = 2)	
long_term_pic_num	ue(v)
if(memory_management_control_operation = = 3 || memory_management_control_operation = = 6)	
long_term_frame_idx	ue(v)
if(memory_management_control_operation = = 4)	
max_long_term_frame_idx_plus1	ue(v)
} while(memory_management_control_operation != 0)	
}	
}	

7.3.3.4 Sample adaptive offset parameter syntax

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000308

sao_param( ) {	C	Descriptor
sample_adaptive_offset_flag	2	u(1)
if (sample_adaptive_offset_flag) {		
sao_split_param(0, 0, 0)		
sao_offset_param(0, 0, 0)		
}		
}		

sao_split_param(x0, y0, saoDepth) {	C	Descriptor
if(saoDepth < SaoMaxDepth)		
sao_split_flag[saoDepth][x0][y0]	2	u(1) | ae(v)
if(sao_split_flag[saoDepth][x0][y0]) {		
sao_split_param(x0 + 0, y0 + 0, saoDepth + 1)		
sao_split_param(x0 + 1, y0 + 0, saoDepth + 1)		
sao_split_param(x0 + 0, y0 + 1, saoDepth + 1)		
sao_split_param(x0 + 1, y0 + 1, saoDepth + 1)		
}		
}		

[Ed.: (WJ) is it better to assign separate section?]

sao_offset_param(x0, y0, saoDepth) {	C	Descriptor
if(sao_split_flag[saoDepth][x0][y0]) {		
sao_offset_param(x0 + 0, y0 + 0, saoDepth + 1)		
sao_offset_param(x0 + 1, y0 + 0, saoDepth + 1)		
sao_offset_param(x0 + 0, y0 + 1, saoDepth + 1)		
sao_offset_param(x0 + 1, y0 + 1, saoDepth + 1)		
} else {		
sao_type_idx[saoDepth][x0][y0]	2	ue(v) | ae(v)
if(sao_type_idx[saoDepth][x0][y0] != 0)		
for(i = 0; i < NumSaoClass; i++)		
sao_offset[saoDepth][x0][y0][i]	2	se(v) | ae(v)
}		
}		

[Ed.: (WJ) is it better to assign separate section?]

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000309

7.3.3.5 Adaptive loop filter parameter syntax

alf_param() {	C	Descriptor
adaptive_loop_filter_flag	2	u(1)
if (adaptive_loop_filter_flag) {		
alf_region_adaptation_flag	2	u(1)
alf_length_luma_minus_5_div2	2	ue(v)
alf_no_filters_minus1	2	ue(v)
if (alf_no_filters_minus1 == 1)		
alf_start_second_filter	2	ue(v)
else if (alf_no_filters_minus1 > 1) {		
for (i=0; i< 16; i++)		
alf_filter_pattern[i]	2	u(1)
}		
if (AlfNumFilters > 1)		
alf_pred_method	2	u(1)
alf_min_kstart_minus1	2	ue(v)
for (i=0; i < AlfMaxDepth; i++)		
alf_golomb_index_bit[i]	2	u(1)
for (i=0; i< AlfNumFilters; i++)		
for (j=0; j< AlfCodedLengthLuma; j++)		
alf_coeff_luma[i][j]		ge(v)
alf_chroma_idc	2	ue(v)
if (alf_chroma_idc) {		
alf_length_chroma_minus_5_div2	2	ue(v)
for(i = 0; i< AlfCodedLengthChroma; i++)		
alf_coeff_chroma[i]		se(v)
}		
}		
}		

7.3.3.6 Adaptive loop filter coding unit control parameter syntax

alf_cu_control_param() {	C	Descriptor
if(adaptive_loop_filter_flag) {		
alf_cu_control_flag	2	u(1)
if(alf_cu_control_flag) {		
alf_cu_control_max_depth	2	ue(v)
~~if(alf_cu_control_max_depth)~~		
~~**alf_length_cu_control_info**~~	2	~~ue~~se(v)
for(i = 0; i < NumAlfCuFlag; i++)		
alf_cu_flag[i]	2	u(1) | ae(v)
}		
}		
}		

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000310

7.3.4 Slice data syntax

slice_data( ) {	Descriptor
CurrTbAddr = first_tb_in_slice[CUAddress]	
moreDataFlag = 1	
if(adaptive_loop_filter_flag && alf_cu_control_flag)	
AlfCuFlagIdx = -1	
do {	
xCU = HorLumaLocation(CurrTbAddr)	
yCU = VerLumaLocation(CurrTbAddr)	
moreDataFlag = coding_tree(xCU, yCU, Log2TbSize)	
if(!entropy_coding_mode_flag)	
moreDataFlag = more_rbsp_data( )	
else {	
end_of_slice_flag	ae(v)
moreDataFlag = !end_of_slice_flag	
}	
CurrTbAddr = NextTbAddress(CurrTbAddr)	
} while(moreDataFlag)	
}	

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000311

7.3.5 Coding tree syntax

coding_tree(x0, y0, log2CUSize) {	Descriptor
if(x0 + (1 << log2CUSize) <= PicWidthInSamples$_L$ && y0 + (1 << log2CUSize) <= PicHeightInSamples$_L$ && ctuAddress(x0, y0) >= SliceAddress) {	
if(!entropy_coding_mode_flag && slice_type != I)	
cu_split_pred_part_mode[x0][y0]	ce(v)
else if(log2CUSize > Log2MinCUSize)	
split_coding_unit_flag[x0 **][** y0 **]**	u(1) | ae(v)
}	
if(adaptive_loop_filter_flag && alf_cu_control_flag) {	
cuDepth = Log2MaxCUSize – log2CUSize	
if(cuDepth <= alf_cu_control_max_depth)	
if(cuDepth == alf_cu_control_max_depth ||	
split_coding_unit_flag[x0][y0] == 0)	
AlfCuFlagIdx++	
}	
if(split_coding_unit_flag[x0][y0]) {	
if(cu_qp_depta_enabled_flag && log2CUSize == log2MinCUDQPSize)	
IsCuQpDeltaCoded = 0	
x1 = x0 + ((1 << log2CUSize) >> 1)	
y1 = y0 + ((1 << log2CUSize) >> 1)	
if(ctuAddress(x1, y0) < SliceAddress)	
moreDataFlag = coding_tree(x0, y0, log2CUSize – 1)	
if(ctuAddress(x0, y1) < SliceAddress && moreDataFlag && x1 < PicWidthInSamples$_L$)	
moreDataFlag = coding_tree(x1, y0, log2CUSize – 1)	
if(ctuAddress(x1, y1) < SliceAddress && moreDataFlag && y1 < PicHeightInSamples$_L$)	
moreDataFlag = coding_tree(x0, y1, log2CUSize – 1)	
if(moreDataFlag && x1 < PicWidthInSamples$_L$ && y1 < PicHeightInSamples$_L$)	
moreDataFlag = coding_tree(x1, y1, log2CUSize – 1)	
} else {	
if(adaptive_loop_filter_flag && alf_cu_control_flag)	
AlfCuFlag[x0][y0] = alf_cu_flag[AlfCuFlagIdx]	
coding_unit(x0, y0, log2CUSize)	
if(!entropy_coding_mode_flag)	
moreDataFlag = more_rbsp_data( )	
else {	
if(granularity_block_boundary(x0, y0, log2CUSize)) {	
end_of_slice_flag	ae(v)
moreDataFlag = !end_of_slice_flag	
} else	
moreDataFlag = 1	
}	
}	
return moreDataFlag	
}	

서식 있음: 강조

서식 있음: 강조

서식 있음: 강조

서식 있음: 강조

서식 있음: (한글) 한국어, (언어2) 영어(미국)

서식 있음: 글꼴: 굵게

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000312

[Ed. (WJ): cuAddress(x, y) returns the address of the smallest coding unit at (x, y), expressed in global decoding order smallest coding unit coordinates, granularity, block_boundary(x0, y0, log2CUSize) returns true when the coding unit specified by (x0, y0, log2CUSize) is the last coding unit in the slice in global decoding order. These functions are not defined yet. They will be added later.]

7.3.6 Coding unit syntax

coding_unit(x0, y0, log2CUSize) {	Descriptor
if(entropy_coding_mode_flag && slice_type != I)	
skip_flag[x0 **][** y0 **]**	u(1) | ae(v)
if(skip_flag[x0][y0])	
prediction_unit(x0, y0, log2CUSize, log2CUSize, 0 , 0)	
else {	
if(!entropy_coding_mode_flag) {	
if(slice_type == I && log2CUSize == Log2MinCUSize)	
intra_part_mode	u(1)
} else if(slice_type != I || log2CUSize == Log2MinCUSize)	
pred_type	u(v) | ae(v)
x1 = x0 + ((1 << log2CUSize) >> 1)	
y1 = y0 + ((1 << log2CUSize) >> 1)	
if(PartMode == PART_2Nx2N) {	
prediction_unit(x0, y0, log2CUSize, log2CUSize, 0 , 0)	
} else if(PartMode == PART_2NxN) {	
prediction_unit(x0, y0, log2CUSize, log2CUSize – 1, 0 , log2CUSize > Log2MinCUSize)	
prediction_unit(x0, y1, log2CUSize, log2CUSize – 1, 1 , 0)	
} else if(PartMode == PART_Nx2N) {	
prediction_unit(x0, y0, log2CUSize - 1, log2CUSize, 0 , log2CUSize > Log2MinCUSize)	
prediction_unit(x1, y0, log2CUSize - 1, log2CUSize, 1 , 0)	
} else { /* PART_NxN */	
prediction_unit(x0, y0, log2CUSize – 1, log2CUSize – 1, 0 , 0)	
prediction_unit(x1, y0, log2CUSize – 1, log2CUSize – 1, 1 , 0)	
prediction_unit(x0, y1, log2CUSize – 1, log2CUSize – 1, 2 , 0)	
prediction_unit(x1, y1, log2CUSize – 1, log2CUSize – 1, 3 , 0)	
}	
}	
if(!pcm_flag) {	
transform_tree(x0, y0, log2CUSize, 0, 0)	
transform_coeff(x0, y0, log2CUSize, 0, 0)	
transform_coeff(x0, y0, log2CUSize, 0, 1)	
transform_coeff(x0, y0, log2CUSize, 0, 2)	
}	
}	

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000313

7.3.7 Prediction unit syntax

prediction_unit(x0, y0, log2PUWidth, log2PUHeight, PartIdx , InferredMergeFlag) {	Descriptor
if(skip_flag[x0][y0]) {	
if(NumMergeCand > 1)	
merge_idx[x0 **][** y0 **]**	ue(v) | ae(v)
} else if(PredMode == MODE_INTRA) {	
if(PartMode == PART_2Nx2N && log2PUWidth >= Log2IPCMCUSize)	
pcm_flag	u(1) | ae(v)
if(pcm_flag) {	
while(!byte_aligned( ))	
pcm_alignment_zero_bit	u(v)
for(i = 0; i < 1 << (log2CUSize << 1); i++)	
pcm_sample_luma[i]	u(v)
for(i = 0; i < (1 << (log2CUSize << 1)) >> 1; i++)	
pcm_sample_chroma[i]	u(v)
} else {	
prev_intra_luma_pred_flag[x0 **][** y0 **]**	u(1) | ae(v)
if(prev_intra_luma_pred_flag[x0][y0])	
if(NumMPMCand > 1)	
mpm_idx[x0 **][** y0 **]**	u(1) | ae(v)
else	
rem_intra_luma_pred_mode[x0 **][** y0 **]**	ce(v) | ae(v)
if(IntraPredMode[x0][y0] == 2)	
planar_flag_luma[x0][y0]	u(1) | ae(v)
intra_chroma_pred_mode[x0][y0]	ue(v) | ae(v)
StandardAsChroma[C] = (chroma_pred_from_luma_enabled_flag ? intra_chroma_pred_mode[x0][y0] == 3 : intra_chroma_pred_mode[x0][y0] == 2)	
if(IntraPredMode[x0][y0] != 2 && IntraPredMode[x0][y0] != 34 && StandardAsChroma[C])	
planar_flag_chroma[x0][y0]	u(1) | ae(v)
}	
} else { /* MODE_INTER */	
if(!InferredMergeFlag)	
if(entropy_coding_mode_flag || PartMode != PART_2Nx2N)	
merge_flag[x0 **][** y0 **]**	u(1) | ae(v)
if(merge_flag[x0][y0] && NumMergeCand > 1) {	
merge_idx[x0 **][** y0 **]**	ue(v) | ae(v)
} else {	
if(slice_type == B) {	
if(!entropy_coding_mode_flag && use_combined_inter_pred_ref(x0, y0)) {	
combined_inter_pred_ref_idx	ue(v)
if(combined_inter_pred_ref_idx == MaxPredRef)	
inter_pred_flag[x0][y0]	ue(v)
} else	
inter_pred_flag[x0 **][** y0 **]**	ue(v) | ae(v)

서식 있음: 글꼴: 굵게
서식 있음: 글꼴: 굵게
서식 있음: 글꼴: 굵게
서식 있음: 글꼴: 굵게
서식 있음: 강조
서식 있음: 글꼴: 굵게
서식 있음: 영어(미국)
서식 있음: 영어(미국)
서식 있음: 강조
서식 있음: 강조
서식 있음: 영어(미국)
서식 있음: 글꼴: 굵게
서식 있음: 영어(미국)
서식 있음: 영어(미국)
서식 있음: 글꼴: 굵게
서식 있음: 글꼴: 굵게
서식 있음: 영어(미국)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000314

}	
if(inter_pred_flag[x0][y0] == Pred_LC) {	
if(num_ref_idx_lc_active_minus1 > 0) {	
if(!entropy_coding_mode_flag && use_combined_inter_pred_ref(x0, y0)) {	
if(combined_inter_pred_ref_idx == MaxPredRef)	
ref_idx_lc_minus4[x0][y0]	ue(v)
} else	
ref_idx_lc[x0 **][** y0 **]**	~~ue(v)~~ ae(v)
}	
mvd_lc[x0 **][** y0 **][** 0 **]**	se(v) | ae(v)
mvd_lc[x0 **][** y0 **][** 1 **]**	se(v) | ae(v)
if(NumMVPCand(LcToLx) > 1)	
mvp_idx_lc[x0][y0]	ue(v) | ae(v)
}	
else { /* Pred_L0 or Pred_BI */	
if(num_ref_idx_l0_active_minus1 > 0) {	
~~**ref_idx_l0[x0][y0]**~~	~~ue(v) | ae(v)~~
if(!entropy_coding_mode_flag && use_combined_inter_pred_ref(x0, y0)) {	
if(combined_inter_pred_ref_idx == MaxPredRef)	
ref_idx_l0_minusX[x0][y0]	ue(v)
} else	
ref_idx_l0_minusX[x0][y0]	ue(v) | ae(v)
}	
mvd_l0[x0 **][** y0 **][** 0 **]**	se(v) | ae(v)
mvd_l0[x0 **][** y0 **][** 1 **]**	se(v) | ae(v)
if(NumMVPCand(L0) > 1)	
mvp_idx_l0[x0][y0]	ue(v) | ae(v)
}	
if(inter_pred_flag[x0][y0] == Pred_BI) {	
if(num_ref_idx_l1_active_minus1 > 0) {	
if(!entropy_coding_mode_flag && use_combined_inter_pred_ref(x0, y0)) {	
if(combined_inter_pred_ref_idx == MaxPredRef)	
ref_idx_l1_minusX[x0][y0]	ue(v)
} else	
ref_idx_l1[x0 **][** y0 **]**	ue(v) | ae(v)
}	
mvd_l1[x0 **][** y0 **][** 0 **]**	se(v) | ae(v)
mvd_l1[x0 **][** y0 **][** 1 **]**	se(v) | ae(v)
if(NumMVPCand(L1) > 1)	
mvp_idx_l1[x0][y0]	ue(v) | ae(v)
}	
}	
}	
}	

서식 있음: 글꼴: 굵게

서식 있음: 영어(미국)

서식 있음: 글꼴: 굵게

서식 있음: 글꼴: 굵게

서식 있음: 글꼴: 굵게

서식 있음: 영어(미국)

[Ed: (BB) NumMergeCand needs to be defined. Decoding processes of neighboring blocks have to be finished to derive NumMergeCand. Same applies for NumMVPCand (TK) and NumMPMCand]

[(TK) Note that the CAVLC syntax may be a little different for the prev_intra_luma_pred_flag[x0][y0] is incorporated into the coding of rem_intra_luma_pred_mode[x0][y0]. It should be decided where the syntax tables for CAVLC (LCEC) should diverge from that of the CABAC.]

[(TK) Note that in CAVLC decoding process, there is also a set of ranking tables and a set of code tables. The selection of these tables are conditional on the NumMPMCand. These tables are available in JCTVC-E131 and should be added to the section for the description of the decoding process for the CAVLC (LCEC) if and when that section becomes available]

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000315

7.3.8 Transform tree syntax

transform_tree(x0, y0, log2TrafoSize, trafoDepth, blkIdx) {	Descriptor
if(entropy_coding_mode_flag && trafoDepth == 0 && IntraSplitFlag == 0) {	
if(PredMode != MODE_INTRA)	
no_residual_data_flag	u(1) | ae(v)
residualDataPresentFlag = !no_residual_data_flag	
} else {	
residualDataPresentFlag = TRUE	
}	
if(residualDataPresentFlag) {	
intraSplitFlag = (IntraSplitFlag && trafoDepth == 0 ? 1 : 0)	
maxDepth = (PredMode == MODE_INTRA ? max_transform_hierarchy_depth_intra + IntraSplitFlag : max_transform_hierarchy_depth_inter)	
if(!entropy_coding_mode_flag) {	
xB = x0 − (x0 & ((1 << log2TrafoSize) − 1))	
yB = y0 − (y0 & ((1 << log2TrafoSize) − 1))	
if(trafoDepth != 0 && !cbf_luma[xB][yB][trafoDepth − 1]) {	
if(cbf_cb[xB][yB][trafoDepth − 1] && log2TrafoSize > Log2MinTrafoSize)	
cbf_cb[x0][y0] [trafoDepth]	u(1)
if(cbf_cr[xB][yB][trafoDepth − 1] && log2TrafoSize > Log2MinTrafoSize)	
cbf_cr[x0][y0] [trafoDepth]	u(1)
if(log2TrafoSize <= Log2MaxTrafoSize && !intraSplitFlag && log2TrafoSize > Log2MinTrafoSize && trafoDepth < maxDepth)	
split_transform_flag[x0][y0] [trafoDepth]	u(1)
} else	
cbp_and_split_transform	vlc(n,v)
}	
if(log2TrafoSize <= Log2MaxTrafoSize && log2TrafoSize > Log2MinTrafoSize && trafoDepth < maxDepth && !intraSplitFlag && entropy_coding_mode_flag)	
split_transform_flag[x0][y0][trafoDepth]	u(1) | ae(v)
if(PredMode != MODE_INTRA && log2TrafoSize <= Log2MaxTrafoSize && entropy_coding_mode_flag) {	
firstChromaCbf = (log2TrafoSize == Log2MaxTrafoSize || trafoDepth == 0 ? 1 : 0)	
if(firstChromaCbf || log2TrafoSize > Log2MinTrafoSize) {	
xBase = x0 − (x0 & ((1 << log2TrafoSize) − 1))	
yBase = y0 − (y0 & ((1 << log2TrafoSize) − 1))	
if(firstChromaCbf || cbf_cb[xBase][yBase][trafoDepth − 1])	
cbf_cb[x0][y0][trafoDepth]	u(1) | ae(v)
if(firstChromaCbf || cbf_cr[xBase][yBase][trafoDepth − 1])	
cbf_cr[x0][y0][trafoDepth]	u(1) | ae(v)
}	
}	
if(split_transform_flag[x0][y0][trafoDepth]) {	
x1 = x0 + ((1 << log2TrafoSize) >> 1)	
y1 = y0 + ((1 << log2TrafoSize) >> 1)	

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000316

transform_tree(x0, y0, log2TrafoSize − 1, trafoDepth + 1, 0)	
transform_tree(x1, y0, log2TrafoSize − 1, trafoDepth + 1, 1)	
transform_tree(x0, y1, log2TrafoSize − 1, trafoDepth + 1, 2)	
transform_tree(x1, y1, log2TrafoSize − 1, trafoDepth + 1, 3)	
} else if(entropy_coding_mode_flag){	
if(PredMode == MODE_INTRA || trafoDepth != 0 || cbf_cb[x0][y0][trafoDepth] || cbf_cr[x0][y0][trafoDepth])	
cbf_luma[x0][y0][trafoDepth]	u(1) | ae(v)
if(PredMode == MODE_INTRA)	
if(log2TrafoSize > Log2MinTrafoSize) {	
cbf_cb[x0][y0][trafoDepth]	u(1) | ae(v)
cbf_cr[x0][y0][trafoDepth]	u(1) | ae(v)
} else if(blkIdx == 0) {	
cbf_cb[x0][y0][trafoDepth − 1]	u(1) | ae(v)
cbf_cr[x0][y0][trafoDepth − 1]	u(1) | ae(v)
}	
}	
}	
}	

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000317

7.3.9 Transform coefficient syntax

transform_coeff(x0, y0, log2TrafoSize, trafoDepth, cIdx) {	Descriptor
if((cIdx = = 0 && cbf_luma[x0][y0][trafoDepth]) || (cIdx = = 1 && cbf_cb[x0][y0][trafoDepth]) || (cIdx = = 2 && cbf_cr[x0][y0][trafoDepth]))) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta	se(v) | ae(v)
IsCuQpDeltaCoded = 1	
}	
if(split_transform_flag[x0][y0][trafoDepth]) {	
x1 = x0 + ((1 << log2TrafoSize) >> 1)	
y1 = y0 + ((1 << log2TrafoSize) >> 1)	
transform_coeff(x0, y0, log2TrafoSize − 1, trafoDepth + 1, cIdx)	
transform_coeff(x1, y0, log2TrafoSize − 1, trafoDepth + 1, cIdx)	
transform_coeff(x0, y1, log2TrafoSize − 1, trafoDepth + 1, cIdx)	
transform_coeff(x1, y1, log2TrafoSize − 1, trafoDepth + 1, cIdx)	
} else {	
if (PredMode = = MODE_INTRA) {	
if (cIdx = = 0)	
scanIdx = ScanType[log2TrafoSize -2][IntraPredMode]	
else	
scanIdx = ScanTypeC[log2TrafoSize -2][IntraPredModeC]	
} else {	
scanIdx = 0	
}	
if (entropy_coding_mode_flag)	
residual_coding_cabac(x0, y0, log2TrafoSize, trafoDepth, scanIdx, cIdx)	
else if (!entropy_coding_mode_flag)	
residual_coding_cavlc(x0, y0, log2TrafoSize, trafoDepth, scanIdx, cIdx)	
}	
}	
}	

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000318

7.3.10 Residual coding CABAC syntax

residual_coding_cabac(x0, y0, log2TrafoSize, trafoDepth, scanIdx, cIdx) {	Descriptor
last_significant_coeff_x	ae(v)
last_significant_coeff_y	ae(v)
n = 0	
xC = ScanOrder[log2TrafoSize − 2][scanIdx][n][0]	
yC = ScanOrder[log2TrafoSize − 2][scanIdx][n][1]	
while((xC != last_significant_coeff_x) || (yC != last_significant_coeff_y)) {	
significant_coeff_flag[xC **][** yC **]**	ae(v)
n++	
xC = ScanOrder[log2TrafoSize − 2][scanIdx][n][0]	
yC = ScanOrder[log2TrafoSize − 2][scanIdx][n][1]	
}	
numSubsets = max(1, (1 << (log2TrafoSize << 1)) >> (cIdx > 0 ? 6 : 4))	
for(i = 0; i < numSubsets; i++) {	
offset = i << 4	
xS = ScanOrder[log2TrafoSize − 2][0][i][0] << 2	
yS = ScanOrder[log2TrafoSize − 2][0][i][1] << 2	
for(n = 0; n < 16; n++) {	
xOffset = n – (n >> 2) << 2	
yOffset = n >> 2	
if(significant_coeff_flag[xS + xOffset][yS + yOffset])	
coeff_abs_level_greater1_flag[n **]**	ae(v)
}	
for(n = 0; n < 16; n++) {	
if(coeff_abs_level_greater1_flag[n]) {	
coeff_abs_level_greater2_flag[n **]**	ae(v)
if(coeff_abs_level_greater2_flag[n])	
coeff_abs_level_minus3[n **]**	ae(v)
}	
}	
for(n = 0; n < 16; n++) {	
xOffset = n – (n >> 2) << 2	
yOffset = n >> 2	
if(significant_coeff_flag[xS + xOffset][yS + yOffset]) {	
coeff_sign_flag[n **]**	ae(v)
transCoeffLevel[x0][y0][trafoDepth][cIdx][n + offset] = (coeff_abs_level_minus3[n] + 3) * (1 − 2 * coeff_sign_flag[n])	
} else	
transCoeffLevel[x0][y0][trafoDepth][cIdx][n + offset] = 0	
}	
}	
}	

7.3.11 Residual coding CAVLC syntax

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000319

residual_coding_cavlc(x0, y0, log2TrafoSize, trafoDepth, cIdx) {	**Descriptor**
n = (1 << (log2TrafoSize << 1)) >> (cIdx > 0 ? 2 : 0)	
last_pos_level_one	ce(v)
lastPos = last_pos_level_one % n	
levelGreaterThanOneFlag = (last_pos_level_one > n)	
level_and_sign(levelGreaterThanOneFlag, transCoeffLevel, x0, y0, trafoDepth, cIdx, lastPos)	
runModeFlag = TRUE	
n = lastPos – 1	
sumBigCoeff = 0	
switchThres = (PredMode == MODE_INTRA && cIdx == 0)? 0 : 49	
while (runModeFlag && n >= 0) {	
run_level_one	ce(v)
runOfZeros = run_level_one % n	
levelGreaterThanOneFlag = (run_level_one > n)	
trOne = (trOne = = 0 | | levelGreaterThanOneFlag) ? 0 : Max(4, trOne + 1)	
n = n – runOfZeros	
if(n >= 0) {	
level_and_sign(levelGreaterThanOneFlag, transCoeffLevel, x0, y0, trafoDepth, cIdx, n)	
if (levelGreaterThanOneFlag) {	
sumBigCoeff = sumBigCoeff + level	
if (log2TrafoSize==2 | | n > switchThres | | sumBigCoeff > 2)	
runModeFlag = FALSE	
}	
}	
n--	
}	
while (n >= 0) {	
level	ce(v)
if (level > 0) {	
sign_flag	u(1)
sign = 1 – 2 * sign_flag	
transCoeffLevel[x0][y0][trafoDepth][cIdx][n] = level * sign	
}	
n--	
}	
}	

7.3.12 Level and sign CAVLC syntax

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000320

level_and_sign(levelGreaterThanOneFlag, transCoeffLevel, x0, y0, trafoDepth, cIdx, n) {	**Descriptor**
if (levelGreaterThanOneFlag) {	
level_minus2_and_sign	ce(v)
level = ((level_minus2_and_sign >> 1) + 2	
sign = 1 - 2*(level_minus2_and_sign & 1)	
transCoeffLevel[x0][y0][trafoDepth][cIdx][n] = level * sign	
}	
else {	
sign_flag	u(1)
transCoeffLevel[x0][y0][trafoDepth][cIdx][n] = 1 – 2 *(sign_flag)	
}	
}	

7.4 Semantics

Semantics associated with the syntax structures and with the syntax elements within these structures are specified in this subclause. When the semantics of a syntax element are specified using a table or a set of tables, any values that are not specified in the table(s) shall not be present in the bitstream unless otherwise specified in this Recommendation | International Standard.

7.4.1 NAL unit semantics

NOTE 1 – The VCL is specified to efficiently represent the content of the video data. The NAL is specified to format that data and provide header information in a manner appropriate for conveyance on a variety of communication channels 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 byte stream is identical except that each NAL unit can be preceded by a start code prefix and extra padding bytes in the byte stream format.

NumBytesInNALunit specifies the size of the NAL unit in bytes. This value is required for decoding of the NAL unit. Some form of demarcation of NAL unit boundaries is necessary to enable inference of NumBytesInNALunit. One such demarcation method is specified in Annex B for the byte stream format. Other methods of demarcation may be specified outside of this Recommendation | International Standard.

forbidden_zero_bit shall be equal to 0.

nal_ref_idc not equal to 0 specifies that the content of the NAL unit contains a sequence parameter set, a picture parameter set or a slice of a reference picture.

For coded video sequences conforming to one or more of the profiles specified in Annex A that are decoded using the decoding process specified in clauses 2-9, nal_ref_idc equal to 0 for a NAL unit containing a slice indicates that the slice is part of a non-reference picture.

nal_ref_idc shall not be equal to 0 for sequence parameter set or picture parameter set NAL units. When nal_ref_idc is equal to 0 for one NAL unit with nal_unit_type equal to 1 or 4 of a particular picture, it shall be equal to 0 for all NAL units with nal_unit_type equal to 1 or 4 of the picture.

nal_ref_idc shall not be equal to 0 for NAL units with nal_unit_type equal to 5.

nal_ref_idc shall be equal to 0 for all NAL units having nal_unit_type equal to 6, 9, 10, 11, or 12.

nal_unit_type specifies the type of RBSP data structure contained in the NAL unit as specified in Table 7-1.

NAL units that use nal_unit_type equal to 0 or in the range of 24..31, inclusive, shall not affect the decoding process specified in this Recommendation | International Standard.

NOTE 2 – NAL unit types 0 and 24..31 may be used as determined by the application. No decoding process for these values of nal_unit_type is specified in this Recommendation | International Standard. Since different applications might use NAL unit types 0 and 24..31 for different purposes, particular care must be exercised in the design of encoders that generate NAL units with nal_unit_type equal to 0 or in the range of 24 to 31, inclusive, and in the design of decoders that interpret the content of NAL units with nal_unit_type equal to 0 or in the range of 24 to 31, inclusive.

Decoders shall ignore (remove from the bitstream and discard) the contents of all NAL units that use reserved values of nal_unit_type.

NOTE 3 – This requirement allows future definition of compatible extensions to this Recommendation | International Standard.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000321

Table 7-13-1-8 – NAL unit type codes, syntax element categories, and NAL unit type classes

nal_unit_type	Content of NAL unit and RBSP syntax structure	NAL unit type class
0	Unspecified	non-VCL
1	Coded slice of a non-IDR and non-CDR picture slice_layer_rbsp( )	VCL
2-3	Reserved	n/a
4	Coded slice of a CDR picture slice_layer_rbsp( )	VCL
5	Coded slice of an IDR picture slice_layer_rbsp( )	VCL
6	Supplemental enhancement information (SEI) sei_rbsp( )	non-VCL
7	Sequence parameter set seq_parameter_set_rbsp( )	non-VCL
8	Picture parameter set pic_parameter_set_rbsp( )	non-VCL
9	Access unit delimiter access_unit_delimiter_rbsp( )	non-VCL
10-11	Reserved	n/a
12	Filler data filler_data_rbsp( )	non-VCL
13-23	Reserved	n/a
24..31	Unspecified	non-VCL

In the text, coded slice NAL unit collectively refers to a coded slice of a non-IDR picture NAL unit or to a coded slice of an IDR picture NAL unit. The variable IdrPicFlag is specified as

$$IdrPicFlag = (nal\_unit\_type == 5) ? 1 : 0 \quad (7\text{-}1)$$

When the value of nal_unit_type is equal to 5 for a NAL unit containing a slice of a particular picture, the picture shall not contain NAL units with nal_unit_type equal to 1 or 4. For coded video sequences conforming to one or more of the profiles specified in Annex A that are decoded using the decoding process specified in clauses 2-9, such a picture is referred to as an IDR picture.

temporal_id specifies a temporal identifier for the NAL unit. The value of temporal_id shall be the same for all NAL units of an access unit. When an access unit contains any NAL unit with nal_unit_type equal to 5, temporal_id shall be equal to 0.

output_flag affects the decoded picture output and removal processes as specified in Annex C.

[Ed. KS: There is no output definition on Annex C yet]

reserved_one_4bits shall be equal to 1. Other values of reserved_one_4bits may be specified in the future by ITU-T | ISO/IEC. Decoders shall ignore the value of reserved_one_4bits.

rbsp_byte[i **]** is the i-th byte of an RBSP. An RBSP is specified as an ordered sequence of bytes as follows.

The RBSP contains an SODB as follows.

- If the SODB is empty (i.e., zero bits in length), the RBSP is also empty.
- Otherwise, the RBSP contains the SODB as follows:
  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:

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000322

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_one_bit equal to 1.

iii) When the rbsp_stop_one_bit is not the last bit of a byte-aligned byte, one or more rbsp_alignment_zero_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 in some RBSPs 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_byte[i] data bytes. The association of the RBSP syntax structures to the NAL units shall be as specified in ~~Table 7-1~~.

NOTE 6 – When 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_one_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_three_byte is a byte equal to 0x03. When an emulation_prevention_three_byte is present in the NAL unit, it shall be discarded by the decoding process.

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

Within the NAL unit, any four-byte sequence that starts with 0x000003 other than the following sequences shall not occur at any byte-aligned position:

– 0x00000300

– 0x00000301

– 0x00000302

– 0x00000303

NOTE 7 – When nal_unit_type is equal to 0, particular care must be exercised in the design of encoders to avoid the presence of the above-listed three-byte and four-byte patterns at the beginning of the NAL unit syntax structure, as the syntax element emulation_prevention_three_byte cannot be the third byte of a NAL unit.

7.4.1.1 Encapsulation of an SODB within an RBSP (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_three_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_one_bit starting at the end of the RBSP,

– 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).

The encoder can produce a NAL unit from an RBSP by the following procedure:

1. 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, when the last byte of the RBSP data is equal to 0x00 (which can only occur when the RBSP ends in a cabac_zero_word), a final byte equal to 0x03 is appended to the end of the data. The last zero byte of a byte-aligned three-byte sequence 0x000000 in the RBSP (which is replaced by the four-byte sequence 0x00000300) is taken into account when searching the RBSP data for the next occurrence of byte-aligned bits with the binary patterns specified above.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000323

2. The resulting sequence of bytes is then prefixed as follows.
   - If nal_unit_type is not equal to 14 or 20, the sequence of bytes is prefixed with the first byte of the NAL unit containing the syntax elements forbidden_zero_bit, nal_ref_idc, and nal_unit_type, where nal_unit_type indicates the type of RBSP data structure the NAL unit contains.
   - Otherwise (nal_unit_type is equal to 14 or 20), the sequence of bytes is prefixed with the first four bytes of the NAL unit, where the first byte contains the syntax elements forbidden_zero_bit, nal_ref_idc, and nal_unit_type and the following three bytes contain the syntax structure nal_unit_header_svc_extension( ). The syntax element nal_unit_type in the first byte indicates the presence of the syntax structure nal_unit_header_svc_extension( ) in the following three bytes and the type of RBSP data structure the NAL unit contains.

The process specified above 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,
- no sequence of 8 zero-valued bits followed by a start code prefix, regardless of byte-alignment, is emulated within the NAL unit.

7.4.1.2 Order of NAL units and association to coded pictures, access units, and video sequences

This subclause specifies constraints on the order of NAL units in the bitstream.

Any order of NAL units in the bitstream obeying these constraints is referred to in the text as the decoding order of NAL units. Within a NAL unit, the syntax in subclauses 7.3, D.1, and E.1 specifies the decoding order of syntax elements. Decoders shall be capable of receiving NAL units and their syntax elements in decoding order.

7.4.1.2.1 Order of sequence and picture parameter set RBSPs and their activation

This subclause specifies the activation process of picture and sequence parameter sets for coded video sequences that conform to one or more of the profiles specified in Annex A that are decoded using the decoding process specified in clauses 2-9.

NOTE 1 – The sequence and picture parameter set mechanism decouples the transmission of infrequently changing information from the transmission of coded macroblock data. Sequence and picture parameter sets may, in some applications, be conveyed "out-of-band" using a reliable transport mechanism.

A picture parameter set RBSP includes parameters that can be referred to by the coded slice NAL units or coded slice data partition A NAL units of one or more coded pictures. Each picture parameter set RBSP is initially considered not active at the start of the operation of the decoding process. At most one picture parameter set RBSP is considered active at any given moment during the operation of the decoding process, and the activation of any particular picture parameter set RBSP results in the deactivation of the previously-active picture parameter set RBSP (if any).

When a picture parameter set RBSP (with a particular value of pic_parameter_set_id) is not active and it is referred to by a coded slice NAL unit or coded slice data partition A NAL unit (using that value of pic_parameter_set_id), it is activated. This picture parameter set RBSP is called the active picture parameter set RBSP until it is deactivated by the activation of another picture parameter set RBSP. A picture parameter set RBSP, with that particular value of pic_parameter_set_id, shall be available to the decoding process prior to its activation.

Any picture parameter set NAL unit containing the value of pic_parameter_set_id for the active picture parameter set RBSP for a coded picture shall have the same content as that of the active picture parameter set RBSP for the coded picture unless it follows the last VCL NAL unit of the coded picture and precedes the first VCL NAL unit of another coded picture.

A sequence parameter set RBSP includes parameters that can be referred to by one or more picture parameter set RBSPs or one or more SEI NAL units containing a buffering period SEI message. Each sequence parameter set RBSP is initially considered not active at the start of the operation of the decoding process. At most one sequence parameter set RBSP is considered active at any given moment during the operation of the decoding process, and the activation of any particular sequence parameter set RBSP results in the deactivation of the previously-active sequence parameter set RBSP (if any).

When a sequence parameter set RBSP (with a particular value of seq_parameter_set_id) is not already active and it is referred to by activation of a picture parameter set RBSP (using that value of seq_parameter_set_id) or is referred to by an SEI NAL unit containing a buffering period SEI message (using that value of seq_parameter_set_id), it is activated. This sequence parameter set RBSP is called the active sequence parameter set RBSP until it is deactivated by the activation of another sequence parameter set RBSP. A sequence parameter set RBSP, with that particular value of seq_parameter_set_id, shall be available to the decoding process prior to its activation. An activated sequence parameter set RBSP shall remain active for the entire coded video sequence.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000324

NOTE 2 – Because an IDR access unit begins a new coded video sequence and an activated sequence parameter set RBSP must remain active for the entire coded video sequence, a sequence parameter set RBSP can only be activated by a buffering period SEI message when the buffering period SEI message is part of an IDR access unit.

Any sequence parameter set NAL unit containing the value of seq_parameter_set_id for the active sequence parameter set RBSP for a coded video sequence shall have the same content as that of the active sequence parameter set RBSP for the coded video sequence unless it follows the last access unit of the coded video sequence and precedes the first VCL NAL unit and the first SEI NAL unit containing a buffering period SEI message (when present) of another coded video sequence.

NOTE 3 – If picture parameter set RBSP or sequence parameter set RBSP are conveyed within the bitstream, these constraints impose an order constraint on the NAL units that contain the picture parameter set RBSP or sequence parameter set RBSP, respectively. Otherwise (picture parameter set RBSP or sequence parameter set RBSP are conveyed by other means not specified in this Recommendation | International Standard), they must be available to the decoding process in a timely fashion such that these constraints are obeyed.

When present, a sequence parameter set extension RBSP includes parameters having a similar function to those of a sequence parameter set RBSP. For purposes of establishing constraints on the syntax elements of the sequence parameter set extension RBSP and for purposes of determining activation of a sequence parameter set extension RBSP, the sequence parameter set extension RBSP shall be considered part of the preceding sequence parameter set RBSP with the same value of seq_parameter_set_id. When a sequence parameter set RBSP is present that is not followed by a sequence parameter set extension RBSP with the same value of seq_parameter_set_id prior to the activation of the sequence parameter set RBSP, the sequence parameter set extension RBSP and its syntax elements shall be considered not present for the active sequence parameter set RBSP.

All constraints that are expressed on the relationship between the values of the syntax elements (and the values of variables derived from those syntax elements) in sequence parameter sets and picture parameter sets and other syntax elements are expressions of constraints that apply only to the active sequence parameter set and the active picture parameter set. If any sequence parameter set RBSP is present that is not activated in the bitstream, its syntax elements shall have values that would conform to the specified constraints if it were activated by reference in an otherwise-conforming bitstream. If any picture parameter set RBSP is present that is not ever activated in the bitstream, its syntax elements shall have values that would conform to the specified constraints if it were activated by reference in an otherwise-conforming bitstream.

During operation of the decoding process (see clause 0), the values of parameters of the active picture parameter set and the active sequence parameter set shall be considered in effect. For interpretation of SEI messages, the values of the parameters of the picture parameter set and sequence parameter set that are active for the operation of the decoding process for the VCL NAL units of the primary coded picture in the same access unit shall be considered in effect unless otherwise specified in the SEI message semantics.

7.4.1.2.2 Order of access units and association to coded video sequences

A bitstream conforming to this Recommendation | International Standard consists of one or more coded video sequences.

A coded video sequence consists of one or more access units. For coded video sequences that conform to one or more of the profiles specified in Annex A that are decoded using the decoding process specified in clauses 2-9, the order of NAL units and coded pictures and their association to access units is described in subclause 7.4.1.2.3.

The first access unit of each coded video sequence is an IDR access unit. All subsequent access units in the coded video sequence are non-IDR access units.

The values of picture order count for the coded pictures in consecutive access units in decoding order containing non-reference pictures shall be non-decreasing.

When present, an access unit following an access unit that contains an end of sequence NAL unit shall be an IDR access unit.

When an SEI NAL unit contains data that pertain to more than one access unit (for example, when the SEI NAL unit has a coded video sequence as its scope), it shall be contained in the first access unit to which it applies.

When an end of stream NAL unit is present in an access unit, this access unit shall be the last access unit in the bitstream and the end of stream NAL unit shall be the last NAL unit in that access unit.

7.4.1.2.3 Order of NAL units and coded pictures and association to access units

This subclause specifies the order of NAL units and coded pictures and association to access unit for coded video sequences that conform to one or more of the profiles specified in Annex A that are decoded using the decoding process specified in clauses 2-9.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000325

An access unit consists of one primary coded picture, zero or more corresponding redundant coded pictures, and zero or more non-VCL NAL units. The association of VCL NAL units to primary or redundant coded pictures is described in subclause 7.4.1.2.5.

The first access unit in the bitstream starts with the first NAL unit of the bitstream.

The first of any of the following NAL units after the last VCL NAL unit of a primary coded picture specifies the start of a new access unit:

- access unit delimiter NAL unit (when present),
- sequence parameter set NAL unit (when present),
- picture parameter set NAL unit (when present),
- SEI NAL unit (when present),
- NAL units with nal_unit_type in the range of 14 to 18, inclusive (when present),
- first VCL NAL unit of a primary coded picture (always present).

The constraints for the detection of the first VCL NAL unit of a primary coded picture are specified in subclause 7.4.1.2.4.

The following constraints shall be obeyed by the order of the coded pictures and non-VCL NAL units within an access unit:

- When an access unit delimiter NAL unit is present, it shall be the first NAL unit. There shall be at most one access unit delimiter NAL unit in any access unit.
- When any SEI NAL units are present, they shall precede the primary coded picture.
- When an SEI NAL unit containing a buffering period SEI message is present, the buffering period SEI message shall be the first SEI message payload of the first SEI NAL unit in the access unit.
- The primary coded picture shall precede the corresponding redundant coded pictures.
- NAL units having nal_unit_type equal to 0, 12, or in the range of 20 to 31, inclusive, shall not precede the first VCL NAL unit of the primary coded picture.

NOTE 1 – Sequence parameter set NAL units or picture parameter set NAL units may be present in an access unit, but cannot follow the last VCL NAL unit of the primary coded picture within the access unit, as this condition would specify the start of a new access unit.

The structure of access units not containing any NAL units with nal_unit_type equal to 0, 7, 8, or in the range of 12 to 18, inclusive, or in the range of 20 to 31, inclusive, is shown in ~~Figure 7-1~~Figure 7-1. [Ed. (TW):adjust text and figure]

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000326

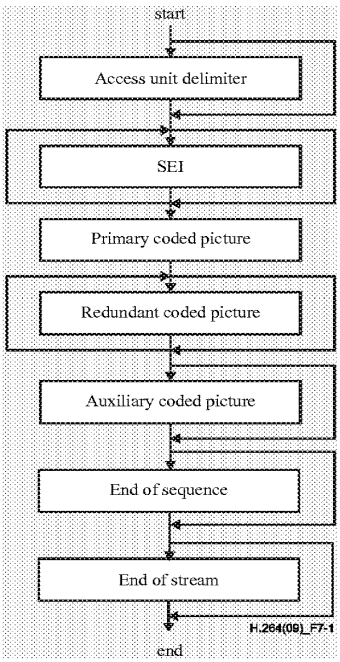


Figure 7-1 – Structure of an access unit not containing any NAL units with nal_unit_type equal to 0, 7, 8, or in the range of 12 to 18, inclusive, or in the range of 20 to 31, inclusive

7.4.1.2.4 Detection of the first VCL NAL unit of a primary coded picture

This subclause specifies constraints on VCL NAL unit syntax that are sufficient to enable the detection of the first VCL NAL unit of each primary coded picture for coded video sequences that conform to one or more of the profiles specified in Annex A that are decoded using the decoding process specified in clauses 2-9.

Any coded slice NAL unit or coded slice data partition A NAL unit of the primary coded picture of the current access unit shall be different from any coded slice NAL unit or coded slice data partition A NAL unit of the primary coded picture of the previous access unit in one or more of the following ways:

– frame_num differs in value. The value of frame_num used to test this condition is the value of frame_num that appears in the syntax of the slice header, regardless of whether that value is inferred to have been equal to 0 for subsequent use in the decoding process due to the presence of memory_management_control_operation equal to 5.

  NOTE 1 – A consequence of the above statement is that a primary coded picture having frame_num equal to 1 cannot contain a memory_management_control_operation equal to 5 unless some other condition listed below is fulfilled for the next primary coded picture that follows after it (if any).

– pic_parameter_set_id differs in value.

– nal_ref_idc differs in value with one of the nal_ref_idc values being equal to 0.

– pic_order_cnt_type is equal to 0 for both and either pic_order_cnt_lsb differs in value, or delta_pic_order_cnt_bottom differs in value.

– pic_order_cnt_type is equal to 1 for both and either delta_pic_order_cnt[0] differs in value, or delta_pic_order_cnt[1] differs in value.

– IdrPicFlag differs in value.

– IdrPicFlag is equal to 1 for both and idr_pic_id differs in value.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000327

7.4.1.2.5 Order of VCL NAL units and association to coded pictures

[Ed. (TW): insert text]

7.4.2 Raw byte sequence payloads and RBSP trailing bits semantics

7.4.2.1 Sequence parameter set RBSP semantics

profile_idc and **level_idc** indicate the profile and level to which the coded video sequence conforms.

reserved_zero_8bits shall be equal to 0. Decoders shall ignore the value of reserved_zero_8bits.

seq_parameter_set_id identifies the sequence parameter set that is referred to by the picture parameter set. The value of seq_parameter_set_id shall be in the range of 0 to 31, inclusive.

max_temporal_layers_minus1 + 1 specifies the maximum number of temporal layers present in the sequence. The value of max_temporal_layers_minus1 shall be in the range of 0 to 7, inclusive.

pic_width_in_luma_samples specifies the width of each decoded picture in luma samples. The value of pic_width_in_luma_samples shall be in the range of 0 to 2^{16}-1, inclusive.

pic_height_in_luma_samples specifies the height of each decoded picture in luma samples. The value of pic_height_in_luma_samples shall be in the range of 0 to 2^{16}-1, inclusive.

bit_depth_luma_minus8 + 8 specifies the bit depth of the samples of the luma array and the value of the luma quantisation parameter range offset $QpBdOffset_Y$, as specified by

$$BitDepth_Y = 8 + bit\_depth\_luma\_minus8 \quad (7\text{-}2)$$
$$QpBdOffset_Y = 6 * bit\_depth\_luma\_minus8 \quad (7\text{-}3)$$

bit_depth_luma_minus8 shall be in the range of 0 to 6, inclusive.

bit_depth_chroma_minus8 + 8 specifies the bit depth of the samples of the chroma arrays and the value of the chroma quantisation parameter range offset $QpBdOffset_C$, as specified by

$$BitDepth_C = 8 + bit\_depth\_chroma\_minus8 \quad (7\text{-}4)$$
$$QpBdOffset_C = 6 * bit\_depth\_chroma\_minus8 \quad (7\text{-}5)$$

bit_depth_chroma_minus8 shall be in the range of 0 to 6, inclusive.

pcm_sample_bit_depth_luma_minus1 + 1 specifies the number of bits used to represent each of PCM sample values of luma component. The value of pcm_sample_bit_depth_luma_minus1 + 1 shall be smaller than or equal to the value of $BitDepth_Y$.

$$PCMBitDepth_Y = 1 + pcm\_sample\_bit\_depth\_luma\_minus1 \quad (7\text{-}64)$$

pcm_sample_bit_depth_chroma_minus1 + 1 specifies the number of bits used to represent each of PCM sample values of chroma components. The value of pcm_sample_bit_depth_chroma_minus1 + 1 shall be smaller than or equal to the value of $BitDepth_C$.

$$PCMBitDepth_C = 1 + pcm\_sample\_bit\_depth\_chroma\_minus1 \quad (7\text{-}74)$$

log2_max_frame_num_minus4 specifies the value of the variable MaxFrameNum that is used in frame_num related derivations as follows:

$$MaxFrameNum = 2^{(\,log2\_max\_frame\_num\_minus4 + 4\,)} \quad (7\text{-}86)$$

The value of log2_max_frame_num_minus4 shall be in the range of 0 to 12, inclusive.

pic_order_cnt_type specifies the method to decode picture order count (as specified in subclause XX). The value of pic_order_cnt_type shall be in the range of 0 to 2, inclusive.

pic_order_cnt_type shall not be equal to 2 in a coded video sequence that contains an access unit containing a non-reference frame followed immediately by an access unit containing a non-reference picture,

log2_max_pic_order_cnt_lsb_minus4 specifies the value of the variable MaxPicOrderCntLsb that is used in the decoding process for picture order count as follows:

서식 있음: 글꼴: 굵게

서식 있음: 아래 첨자

서식 있음: 아래 첨자

서식 있음: 들여쓰기: 왼쪽: 1.27 cm, 간격 앞: 9.65 pt, 단락 뒤: 12 pt, 탭: 24.25 글자, 가운데 + 48.5 글자, 오른쪽

서식 있음: 글꼴: 굵게

서식 있음: 아래 첨자

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000328

$$MaxPicOrderCntLsb = 2^{(\ log2\_max\_pic\_order\_cnt\_lsb\_minus4 + 4\ )} \quad (7\text{-}97)$$

The value of log2_max_pic_order_cnt_lsb_minus4 shall be in the range of 0 to 12, inclusive.

delta_pic_order_always_zero_flag equal to 1 specifies that delta_pic_order_cnt[0] and delta_pic_order_cnt[1] are not present in the slice headers of the sequence and shall be inferred to be equal to 0. delta_pic_order_always_zero_flag equal to 0 specifies that delta_pic_order_cnt[0] is present in the slice headers of the sequence and delta_pic_order_cnt[1] may be present in the slice headers of the sequence.

offset_for_non_ref_pic is used to calculate the picture order count of a non-reference picture as specified in subclause XX. The value of offset_for_non_ref_pic shall be in the range of $-2^{31}+1$ to $2^{31}-1$, inclusive.

num_ref_frames_in_pic_order_cnt_cycle is used in the decoding process for picture order count as specified in subclause XX. The value of num_ref_frames_in_pic_order_cnt_cycle shall be in the range of 0 to 255, inclusive.

offset_for_ref_frame[i] is an element of a list of num_ref_frames_in_pic_order_cnt_cycle values used in the decoding process for picture order count as specified in subclause XX. The value of offset_for_ref_frame[i] shall be in the range of $-2^{31}+1$ to $2^{31}-1$, inclusive.

When pic_order_cnt_type is equal to 1, the variable ExpectedDeltaPerPicOrderCntCycle is derived by

```
ExpectedDeltaPerPicOrderCntCycle = 0
for( i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++ )
    ExpectedDeltaPerPicOrderCntCycle += offset_for_ref_frame[ i ]
```
(7-108)

max_num_ref_frames specifies the maximum number of short-term and long-term reference frames, complementary reference field pairs, and non-paired reference fields that may be used by the decoding process for inter prediction of any picture in the sequence. max_num_ref_frames also determines the size of the sliding window operation. The value of max_num_ref_frames shall be in the range of 0 to MaxDpbFrames, inclusive.

gaps_in_frame_num_value_allowed_flag specifies the allowed values of frame_num and the decoding process in case of an inferred gap between values of frame_num.

log2_min_coding_block_size_minus3 specifies the minimum size of a coding block.

The variable Log2MinCUSize is set equal to log2_min_coding_block_size_minus3 + 3.

log2_diff_max_min_coding_block_size specifies the difference between the maximum and minimum coding block size.

The variable Log2MaxCUSize is set equal to log2_min_coding_block_size_minus 3 + 3 + log2_diff_max_min_coding_block_size.

log2_min_transform_block_size_minus2 specifies the minimum size of a transform block.

The variable Log2MinTrafoSize is set equal to log2_min_transform_block_size_minus2 + 2.

log2_diff_max_min_transform_block_size specifies the difference between the maximum and minimum transform size.

The variable Log2MaxTrafoSize is set equal to log2_min_transform_block_size_minus 2 + 2 + log2_diff_max_min_transform_block_size.

The bitstream shall not contain data that result in Log2MaxTrafoSize greater than Log2MaxCUSize.

log2_min_pcm_coding_block_size_minus3 + 3 specifies the minimum size of I_PCM coding blocks.

The variable Log2MinIPCMCUSize is set equal to log2_min_pcm_coding_block_size_minus3 + 3. The variable Log2MinIPCMCUSize should be equal or less than Log2MaxCUSize.

max_transform_hierarchy_depth_intra specifies the maximum hierarchy depth for transform blocks of coding blocks coded in intra prediction mode.

max_transform_hierarchy_depth_inter specifies the maximum hierarchy depth for transform units of coding units coded in inter prediction mode.

chroma_pred_from_luma_enabled_flag equal to 1 specifies the intra chroma prediction process using the reconstructed luma block is applied according to the intra chroma prediction mode.

loop_filter_across_slice_flag equal to 1 specifies the in-loop filtering operations are performed across slice boundary; otherwise, the in-loop operations are slice-independent and not across slice boundary. The in-loop filtering operations include deblocking filter, sample adaptive offset, and adaptive loop filter.

서식 있음: 글꼴: 굵게

서식 있음: 글꼴: 굵게

서식 있음: 글꼴: 굵게

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000329

sample_adaptive_offset_enabled_flag equal to 1 specifies the sample adaptive offset process is applied to the reconstruced picture after the deblocking filter process.

adaptive_loop_filter_enabled_flag equal to 1 specifies the adaptive loop filter process is applied to the reconstructed picture after the deblocking filter process.

서식 있음: 글꼴: 굵게

pcm_loop_filter_disable_flag specifies whether the loop filter process is disabled on reconstructed pixels of I_PCM blocks. If the pcm_loop_filter_disable_flag value is equal to 1, both deblocking and adaptive loop filter processes on the reconstructed pixels of I_PCM blocks are disabled; otherwise, the pcm_loop_filter_disable_flag value is equal to 0, both deblocking and adaptive loop filter processes on the reconstructed pixels of I_PCM blocks are not disabled.

[Ed. (WJ): select one expression – enabled_flag or disable_flag]

cu_qp_delta_enabled_flag equal to 1 specifies cu_qp_delta presents in coding unit layer to further modify the value of QP_Y.

temporal_id_nesting_flag specifies whether inter prediction is additionally restricted for the coded video sequence.

Dependent on temporal_id_nesting_flag, the following applies.

– If temporal_id_nesting_flag is equal to 0, additional constraints may not be obeyed.

– Otherwise (temporal_id_nesting_flag is equal to 1), the following applies.

  – For each access unit auA with temporal_id equal to tIdA, an access unit auB with temporal_id equal to tIdB and tIdB less than or equal to tIdA shall not be referenced by inter prediction when there exists an access unit auC with temporal_id equal to tIdC and tIdC less than tIdB, which follows the access unit auB and precedes the access unit auA in decoding order.

  NOTE – The syntax element temporal_id_nesting_flag is used to indicate that temporal up-switching, i.e., switching from decoding of up to a specific temporal_id tIdN to decoding up to a temporal_id tIdM > tIdN, is always possible.

7.4.2.2 Picture parameter set RBSP semantics

pic_parameter_set_id identifies the picture parameter set that is referred to in the slice header. The value of pic_parameter_set_id shall be in the range of 0 to 255, inclusive.

seq_parameter_set_id refers to the active sequence parameter set. The value of seq_parameter_set_id shall be in the range of 0 to 31, inclusive.

entropy_coding_mode_flag selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables as follows.

– If entropy_coding_mode_flag is equal to 0, the method specified by the left descriptor in the syntax table is applied.

– Otherwise (entropy_coding_mode_flag is equal to 1), the method specified by the right descriptor in the syntax table is applied.

num_temporal_layer_switching_point_flags specifies how many temporal switching point flags are present. If temporal_id_nesting_flag is equal to 1, num_temporal_layer_switching_point_flags shall be equal to 0.

temporal_layer_switching_point_flag[i] specifies if the current access point is a temporal switching point allowing the decoding of higher temporal id layers following this access unit. If temporal_id_nesting_flag is equal to 1, temporal_layer_switching_point_flag[i] shall be inferred to be equal to 1. If temporal_id_nesting_flag is equal to 0 and num_temporal_layer_switching_point_flags is less than i, temporal_layer_switching_point_flag[i] shall be inferred to be equal to 0 .

NOTE – When starting to decode a higher temporal layer i, availability of required reference pictures can be guaranteed immediately following an IDR, or a picture with the temporal_id value j lower than i and temporal_switching_flag[j] equal to 1.

num_ref_idx_l0_default_active_minus1 specifies how num_ref_idx_l0_active_minus1 is inferred for P and B slices with num_ref_idx_active_override_flag equal to 0. The value of num_ref_idx_l0_default_active_minus1 shall be in the range of 0 to 31, inclusive.

num_ref_idx_l1_default_active_minus1 specifies how num_ref_idx_l1_active_minus1 is inferred for B slices with num_ref_idx_active_override_flag equal to 0. The value of num_ref_idx_l1_default_active_minus1 shall be in the range of 0 to 31, inclusive.

pic_init_qp_minus26 specifies the initial value minus 26 of $SliceQP_Y$ 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 cu_qp_delta is decoded at the coding unit layer. The value of pic_init_qp_minus26 shall be in the range of $-(26 + QpBdOffset_Y)$ to +25, inclusive.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000330

constrained_intra_pred_flag equal to 0 specifies that intra prediction allows usage of residual data and decoded samples of neighbouring macroblocks coded using Inter macroblock prediction modes for the prediction of macroblocks coded using Intra macroblock prediction modes. constrained_intra_pred_flag equal to 1 specifies constrained intra prediction, in which case prediction of macroblocks coded using Intra macroblock prediction modes only uses residual data and decoded samples from I macroblock types.

slice_granularity indicates the slice granularity within a picture. The value of slice_granularity shall not be larger than Min(Log2MaxCUSize – 4, log2_diff_max_min_coding_block_size). The variable SliceGranularity is set to the value of (slice_granularity << 1).

shared_pps_info_enabled_flag specifies the shared information in picture parameter set RBSP shall be used for the referred slices. If shared_pps_info_enabled_flag is equal to 1, the alf_param() in picture parameter set RBSP shall be applied for the referred slices; otherwise, the alf_param() in slice header(s) shall be applied.

max_cu_qp_delta_depth specifies the maximum allowed depth that is used for specifying QP_Y values for coding unit. The value of max_cu_qp_delta_depth shall be in the range of 0 to 15, inclusive.

The variable log2MinCUDQPSize specifying the minimum coding unit size that can further modifies the value of QP_Y as follows:

$$\text{log2MinCUDQPSize} = \text{Log2MaxCUSize} - \text{max\_cu\_qp\_delta\_depth} \quad (7\text{-}117)$$

7.4.2.3 Supplemental enhancement information RBSP semantics

Supplemental Enhancement Information (SEI) contains information that is not necessary to decode the samples of coded pictures from VCL NAL units.

7.4.2.3.1 Supplemental enhancement information message semantics

An SEI RBSP contains one or more SEI messages. Each SEI message consists of the variables specifying the type payloadType and size payloadSize of the SEI payload. SEI payloads are specified in Annex D. The derived SEI payload size payloadSize is specified in bytes and shall be equal to the number of RBSP bytes in the SEI payload.

NOTE – The NAL unit byte sequence containing the SEI message might include one or more emulation prevention bytes (represented by emulation_prevention_three_byte syntax elements). Since the payload size of an SEI message is specified in RBSP bytes, the quantity of emulation prevention bytes is not included in the size payloadSize of an SEI payload.

ff_byte is a byte equal to 0xFF identifying a need for a longer representation of the syntax structure that it is used within.

last_payload_type_byte is the last byte of the payload type of an SEI message.

last_payload_size_byte is the last byte of the payload size of an SEI message.

7.4.2.4 Access unit delimiter RBSP semantics

The access unit delimiter may be used to indicate the type of slices present in a primary coded picture and to simplify the detection of the boundary between access units. There is no normative decoding process associated with the access unit delimiter.

primary_pic_type indicates that the slice_type values for all slices of the primary coded picture are members of the set listed in Table 7-2~~Table 7-2~~ for the given value of primary_pic_type.

Table 7-2~~7-3~~ – Meaning of primary_pic_type

primary_pic_type	slice_type values that may be present in the primary coded picture
0	2
1	0, 2
2	0, 1, 2

7.4.2.5 Filler data RBSP semantics

The filler data RBSP contains bytes whose value shall be equal to 0xFF. No normative decoding process is specified for a filler data RBSP.

ff_byte is a byte equal to 0xFF.

서식 있음: 글꼴: 굵게

서식 있음: 제목 5

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000331

7.4.2.6 Slice layer RBSP semantics

The slice layer RBSP consists of a slice header and slice data.

[Ed. (TW): insert text]

7.4.2.7 RBSP slice trailing bits semantics

cabac_zero_word is a byte-aligned sequence of two bytes equal to 0x0000.

Let NumBytesInVclNALunits be the sum of the values of NumBytesInNALunit for all VCL NAL units of a coded picture.

Let BinCountsInNALunits be the number of times that the parsing process function DecodeBin( ), specified in subclause 9.3.3.2, is invoked to decode the contents of all VCL NAL units of a coded picture. When entropy_coding_mode_flag is equal to 1, BinCountsInNALunits shall not exceed (32 ÷ 3) * NumBytesInVclNALunits + (RawMbBits * PicSizeInMbs) ÷ 32.

NOTE – The constraint on the maximum number of bins resulting from decoding the contents of the slice layer NAL units can be met by inserting a number of cabac_zero_word syntax elements to increase the value of NumBytesInVclNALunits. Each cabac_zero_word is represented in a 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_three_byte for each cabac_zero_word).

7.4.2.8 RBSP trailing bits semantics

rbsp_stop_one_bit shall be equal to 1.

rbsp_alignment_zero_bit shall be equal to 0.

7.4.3 Slice header semantics

When present, the value of the slice header syntax elements pic_parameter_set_id, frame_num, field_pic_flag, bottom_field_flag, idr_pic_id, pic_order_cnt_lsb, delta_pic_order_cnt[0], delta_pic_order_cnt[1], and slice_group_change_cycle shall be the same in all slice headers of a coded picture.

~~first_tb_in_slice specifies the address of the first treeblock in the slice.~~

~~The first treeblock address of the slice is derived as follows.~~

~~TBD.~~

~~entropy~~lightweight_slice_flag ~~equal to 1 specifies the slice is an entropy slice. The entropy slice denotes that the entropy decoding process shall use an alternative definition of neighbor availability than the reconstruction process whereas entropy_slice_flag equal to 0 denotes that the entropy decoding and reconstruction processes shall use the same definition of neighbor availability. [Ed.: (WJ) needs to be clarified more]~~equal to 1 specifies that the value of slice header syntax elements not present shall be inferred to be equal to the value of slice header syntax elements in a proceeding slice, where a proceeding slice is defined as the slice containing treeblock with location (i, CTUAddr[first_tb_in_slice – 1]). lightweight_slice_flag shall be equal to 0 when first_tb_in_slice[CTUAddr] equal to 0. [Ed.: (WJ): do we need to code lightweight_slice_flag when first_tb_in_slice is equal to 0?]

slice_type specifies the coding type of the slice according to Table 7-3~~Table 7-3~~.

Table 7-3~~7-3~~ – Name association to slice_type

slice_type	Name of slice_type
0	P (P slice)
1	B (B slice)
2	I (I slice)

When nal_unit_type is equal to 5 (IDR picture), slice_type shall be equal to 2.

When max_num_ref_frames is equal to 0, slice_type shall be equal to 2.

pic_parameter_set_id specifies the picture parameter set in use. The value of pic_parameter_set_id shall be in the range of 0 to 255, inclusive.

~~frame~~**_num** is used as an identifier for pictures and shall be represented by log2_max_~~frame~~_num_minus4 + 4 bits in the bitstream. frame_num is constrained as follows:

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000332

The variable PrevRefFrameNum is derived as follows.

- If the current picture is an IDR picture, PrevRefFrameNum is set equal to 0.
- Otherwise (the current picture is not an IDR picture), PrevRefFrameNum is set as follows.
  - If the decoding process for gaps in frame_num was invoked by the decoding process for an access unit that contained a non-reference picture that followed the previous access unit in decoding order that contained a reference picture, PrevRefFrameNum is set equal to the value of frame_num for the last of the "non-existing" reference frames inferred by the decoding process for gaps in frame_num.
  - Otherwise, PrevRefFrameNum is set equal to the value of frame_num for the previous access unit in decoding order that contained a reference picture.

The value of frame_num is constrained as follows.

- If the current picture is an IDR picture, frame_num shall be equal to 0.
- Otherwise (the current picture is not an IDR picture), referring to the primary coded picture in the previous access unit in decoding order that contains a reference picture as the preceding reference picture, the value of frame_num for the current picture shall not be equal to PrevRefFrameNum unless both of the following conditions are true.
  - the current picture and the preceding reference picture belong to consecutive access units in decoding order
  - one or more of the following conditions is true
    - the preceding reference picture is an IDR picture
    - the preceding reference picture includes a memory_management_control_operation syntax element equal to 5

      NOTE 1 – When the preceding reference picture includes a memory_management_control_operation syntax element equal to 5, PrevRefFrameNum is equal to 0.
    - there is a primary coded picture that precedes the preceding reference picture and the primary coded picture that precedes the preceding reference picture does not have frame_num equal to PrevRefFrameNum
    - there is a primary coded picture that precedes the preceding reference picture and the primary coded picture that precedes the preceding reference picture is not a reference picture

When the value of frame_num is not equal to PrevRefFrameNum, the following applies.

- There shall not be any previous frame in decoding order that is currently marked as "used for short-term reference" that has a value of frame_num equal to any value taken on by the variable UnusedShortTermFrameNum in the following:

```
UnusedShortTermFrameNum = ( PrevRefFrameNum + 1 ) % MaxFrameNum
while( UnusedShortTermFrameNum  !=  frame_num )
    UnusedShortTermFrameNum = ( UnusedShortTermFrameNum + 1 ) % MaxFrameNum
```

- The value of frame_num is constrained as follows.
  - If gaps_in_frame_num_value_allowed_flag is equal to 0, the value of frame_num for the current picture shall be equal to (PrevRefFrameNum + 1) % MaxFrameNum.
  - Otherwise (gaps_in_frame_num_value_allowed_flag is equal to 1), the following applies.
    - If frame_num is greater than PrevRefFrameNum, there shall not be any non-reference pictures in the bitstream that follow the previous reference picture and precede the current picture in decoding order in which either of the following conditions is true.
      - The value of frame_num for the non-reference picture is less than PrevRefFrameNum.
      - The value of frame_num for the non-reference picture is greater than the value of frame_num for the current picture.
    - Otherwise (frame_num is less than PrevRefFrameNum), there shall not be any non-reference pictures in the bitstream that follow the previous reference picture and precede the current picture in decoding order in which both of the following conditions are true.
      - The value of frame_num for the non-reference picture is less than PrevRefFrameNum.
      - The value of frame_num for the non-reference picture is greater than the value of frame_num for the current picture.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000333

A picture including a memory_management_control_operation equal to 5 shall have frame_num constraints as described above and, after the decoding of the current picture and the processing of the memory management control operations, the picture shall be inferred to have had frame_num equal to 0 for all subsequent use in the decoding process.

idr_pic_id identifies an IDR picture. The values of idr_pic_id in all the slices of an IDR picture shall remain unchanged. When two consecutive access units in decoding order are both IDR access units, the value of idr_pic_id in the slices of the first such IDR access unit shall differ from the idr_pic_id in the second such IDR access unit. The value of idr_pic_id shall be in the range of 0 to 65535, inclusive.

pic_order_cnt_lsb specifies the picture order count modulo MaxPicOrderCntLsb for the top field of a coded frame or for a coded field. The length of the pic_order_cnt_lsb syntax element is log2_max_pic_order_cnt_lsb_minus4 + 4 bits. The value of the pic_order_cnt_lsb shall be in the range of 0 to MaxPicOrderCntLsb − 1, inclusive.

num_ref_idx_active_override_flag equal to 1 specifies that the syntax element num_ref_idx_l0_active_minus1 is present for P and B slices and that the syntax element num_ref_idx_l1_active_minus1 is present for B slices. num_ref_idx_active_override_flag equal to 0 specifies that the syntax elements num_ref_idx_l0_active_minus1 and num_ref_idx_l1_active_minus1 are not present.

When the current slice is a P or B slice and field_pic_flag is equal to 0 and the value of num_ref_idx_l0_default_active_minus1 in the picture parameter set exceeds 15, num_ref_idx_active_override_flag shall be equal to 1.

When the current slice is a B slice and field_pic_flag is equal to 0 and the value of num_ref_idx_l1_default_active_minus1 in the picture parameter set exceeds 15, num_ref_idx_active_override_flag shall be equal to 1.

num_ref_idx_l0_active_minus1 specifies the maximum reference index for reference picture list 0 that shall be used to decode the slice.

When the current slice is a P or B slice and num_ref_idx_l0_active_minus1 is not present, num_ref_idx_l0_active_minus1 shall be inferred to be equal to num_ref_idx_l0_default_active_minus1.

The range of num_ref_idx_l0_active_minus1 is specified as follows.p

- If field_pic_flag is equal to 0, num_ref_idx_l0_active_minus1 shall be in the range of 0 to 15, inclusive. When MbaffFrameFlag is equal to 1, num_ref_idx_l0_active_minus1 is the maximum index value for the decoding of frame macroblocks and 2 * num_ref_idx_l0_active_minus1 + 1 is the maximum index value for the decoding of field macroblocks.
- Otherwise (field_pic_flag is equal to 1), num_ref_idx_l0_active_minus1 shall be in the range of 0 to 31, inclusive.

num_ref_idx_l1_active_minus1 specifies the maximum reference index for reference picture list 1 that shall be used to decode the slice.

When the current slice is a B slice and num_ref_idx_l1_active_minus1 is not present, num_ref_idx_l1_active_minus1 shall be inferred to be equal to num_ref_idx_l1_default_active_minus1.

The range of num_ref_idx_l1_active_minus1 is constrained as specified in the semantics for num_ref_idx_l0_active_minus1 with l0 and list 0 replaced by l1 and list 1, respectively.

cabac_init_idc specifies the index for determining the initialisation table used in the initialisation process for context variables. The value of cabac_init_idc shall be in the range of 0 to 2, inclusive.

first_slice_in_pic_flag indicates whether the slice is the first slice of the picture. If first_slice_in_pic_flag is equal to 1, the variables SliceAddress and LCUAddress are both set to 0 and the decoding starts with the first LCU in the picture.

slice_address specifies the address in slice granularity resolution in which the slice starts and shall be represented by Ceil(Log2(NumLCUsInPicture)) + SliceGranularity) bits in the bitstream where NumLCUsInPicture is the number of LCUs in a picture.

The variable LCUAddress is set to (slice_address >> SliceGranularity) and represents the LCU part of the slice address in raster scan order. The variable GranularityAddress is set to (slice_address - (LCUAddress << SliceGranularity)) and represents the sub-LCU part of the slice address expressed in z-scan order.

The variable SliceAddress is then set to (LCUAddress << (log2_diff_max_min_coding_block_size << 1)) + (GranularityAddress << ((log2_diff_max_min_coding_block_size << 1) - SliceGranularity)) and the slice decoding starts with the largest coding unit possible at the slice starting coordinate.

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 cu_qp_delta in the coding unit layer. The initial QP_Y quantisation parameter for the slice is computed as

$$SliceQP_Y = 26 + pic\_init\_qp\_minus26 + slice\_qp\_delta \quad (7\text{-}129)$$

서식 있음: 글꼴: 굵게

서식 있음: 글꼴: 굵게

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000334

The value of slice_qp_delta shall be limited such that $SliceQP_Y$ is in the range of $-QpBdOffset_Y$ to +51, inclusive.

collocated_from_l0_flag equal to 1 specifies the picture that contains the co-located partition shall be derived from list 0, otherwise the picture shall be derived from list 1.

7.4.3.1 Reference picture list modification semantics

The syntax elements modification_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_num specify the change from the initial reference picture lists to the reference picture lists to be used for decoding the slice.

ref_pic_list_modification_flag_l0 equal to 1 specifies that the syntax element modification_of_pic_nums_idc is present for specifying reference picture list 0. ref_pic_list_modification_flag_l0 equal to 0 specifies that this syntax element is not present.

When ref_pic_list_modification_flag_l0 is equal to 1, the number of times that modification_of_pic_nums_idc is not equal to 3 following ref_pic_list_modification_flag_l0 shall not exceed num_ref_idx_l0_active_minus1 + 1.

When RefPicList0[num_ref_idx_l0_active_minus1] in the initial reference picture list produced as specified in subclause 8.2.2.2 is equal to "no reference picture", ref_pic_list_modification_flag_l0 shall be equal to 1 and modification_of_pic_nums_idc shall not be equal to 3 until RefPicList0[num_ref_idx_l0_active_minus1] in the modified list produced as specified in subclause 8.2.2.3 is not equal to "no reference picture".

ref_pic_list_modification_flag_l1 equal to 1 specifies that the syntax element modification_of_pic_nums_idc is present for specifying reference picture list 1. ref_pic_list_modification_flag_l1 equal to 0 specifies that this syntax element is not present.

When ref_pic_list_modification_flag_l1 is equal to 1, the number of times that modification_of_pic_nums_idc is not equal to 3 following ref_pic_list_modification_flag_l1 shall not exceed num_ref_idx_l1_active_minus1 + 1.

When decoding a slice with slice_type equal to 1 or 6 and RefPicList1[num_ref_idx_l1_active_minus1] in the initial reference picture list produced as specified in subclause 8.2.2.2 is equal to "no reference picture", ref_pic_list_modification_flag_l1 shall be equal to 1 and modification_of_pic_nums_idc shall not be equal to 3 until RefPicList1[num_ref_idx_l1_active_minus1] in the modified list produced as specified in subclause 8.2.2.3 is not equal to "no reference picture".

modification_of_pic_nums_idc together with abs_diff_pic_num_minus1 or long_term_pic_num specifies which of the reference pictures are re-mapped. The values of modification_of_pic_nums_idc are specified in Table 7-4~~Table 7-4~~. The value of the first modification_of_pic_nums_idc that follows immediately after ref_pic_list_modification_flag_l0 or ref_pic_list_modification_flag_l1 shall not be equal to 3.

Table 7-4~~7-4~~ – modification_of_pic_nums_idc operations for modification of reference picture lists

modification_of_pic_nums_idc	modification specified
0	abs_diff_pic_num_minus1 is present and corresponds to a difference to subtract from a picture number prediction value
1	abs_diff_pic_num_minus1 is present and corresponds to a difference to add to a picture number prediction value
2	long_term_pic_num is present and specifies the long-term picture number for a reference picture
3	End loop for modification of the initial reference picture list

abs_diff_pic_num_minus1 plus 1 specifies the absolute difference between the picture number of the picture being moved to the current index in the list and the picture number prediction value. abs_diff_pic_num_minus1 shall be in the range of 0 to MaxPicNum − 1. The allowed values of abs_diff_pic_num_minus1 are further restricted as specified in subclause 8.2.2.3.1.

[Ed. (TW): clarify the following paragraph]

long_term_pic_num specifies the long-term picture number of the picture being moved to the current index in the list. When decoding a coded frame, long_term_pic_num shall be equal to a LongTermPicNum assigned to one of the reference frames or complementary reference field pairs marked as "used for long-term reference". When decoding a coded field, long_term_pic_num shall be equal to a LongTermPicNum assigned to one of the reference fields marked as "used for long-term reference".

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000335

7.4.3.2 Reference picture lists combination semantics

ref_pic_list_combination_flag equal to 1 indicates that the reference picture list 0 and the reference picture list 1 are combined to be an additional reference picture lists combination used for the prediction units being uni-directional predicted. This flag equal to 0 indicates that the reference picture list 0 and reference picture list 1 are identical thus reference picture list 0 is used as the reference picture lists combination. The reference picture lists combination is set to be empty at the start of the loop defined in this table.

num_ref_idx lc_active_minus1+1 specifies the number of reference pictures selected from reference picture list 0 or reference picture list 1 in the reference picture lists combination.

ref_pic_list_modification_flag_lc equal to 1 specifies that the syntax element pic_from_list_0_flag and ref_idx_list_curr are present for specifying the mapping for the entries of the reference picture lists combination to the entries of reference picture list 0 and reference picture list 1.

ref_pic_list_modification_flag_lc equal to 0 specifies that these syntax elements are not present. The reference picture lists combination is initialized as specified in subsclause X.X.X.X.

pic_from_list_0_flag indicates the current reference picture added into the reference picture lists combination is from reference picture list 0 or reference picture list 1. When this flag is equal to 1, the picture is from the reference picture list 0, and the CurrRefPicList is reference picture list 0; when this flag is equal to 0, the picture is from the reference picture list 1, and the CurrRefPicList is reference picture list 1;

ref_idx_list_curr indicates the reference index of the picture in the CurrRefPicList to be appended at the end of the reference picture lists combination.

7.4.3.3 Decoded reference picture marking semantics

[Ed. (TW): clarify the following paragraphs wrt frame/field/picture]

The syntax elements no_output_of_prior_pics_flag, long_term_reference_flag, adaptive_ref_pic_marking_mode_flag, memory_management_control_operation, difference_of_pic_nums_minus1, long_term_frame_idx, long_term_pic_num, and max_long_term_frame_idx_plus1 specify marking of the reference pictures.

The marking of a reference picture can be "unused for reference", "used for short-term reference", or "used for long-term reference", but only one among these three. When a reference picture is referred to as being marked as "used for reference", this collectively refers to the picture being marked as "used for short-term reference" or "used for long-term reference" (but not both). A reference picture that is marked as "used for short-term reference" is referred to as a short-term reference picture. A reference picture that is marked as "used for long-term reference" is referred to as a long-term reference picture.

The content of the decoded reference picture marking syntax structure shall be the same in all slice headers of the primary coded picture.

The syntax category of the decoded reference picture marking syntax structure shall be inferred as follows.

- If the decoded reference picture marking syntax structure is in a slice header, the syntax category of the decoded reference picture marking syntax structure is inferred to be equal to 2.
- Otherwise (the decoded reference picture marking syntax structure is in a decoded reference picture marking repetition SEI message as specified in Annex D), the syntax category of the decoded reference picture marking syntax structure is inferred to be equal to 5.

no_output_of_prior_pics_flag specifies how the previously-decoded pictures in the decoded picture buffer are treated after decoding of an IDR picture. When the IDR picture is the first IDR picture in the bitstream, the value of no_output_of_prior_pics_flag has no effect on the decoding process. When the IDR picture is not the first IDR picture in the bitstream and the value of PicWidthInMbs, FrameHeightInMbs, or max_dec_frame_buffering derived from the active sequence parameter set is different from the value of PicWidthInMbs, FrameHeightInMbs, or max_dec_frame_buffering derived from the sequence parameter set active for the preceding picture, no_output_of_prior_pics_flag equal to 1 may (but should not) be inferred by the decoder, regardless of the actual value of no_output_of_prior_pics_flag.

long_term_reference_flag equal to 0 specifies that the MaxLongTermFrameIdx variable is set equal to "no long-term picture indices" and that the IDR picture is marked as "used for short-term reference". long_term_reference_flag equal to 1 specifies that the MaxLongTermFrameIdx variable is set equal to 0 and that the current IDR picture is marked "used for long-term reference" and is assigned LongTermFrameIdx equal to 0. When max_num_ref_frames is equal to 0, long_term_reference_flag shall be equal to 0.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000336

adaptive_ref_pic_marking_mode_flag selects the reference picture marking mode of the currently decoded picture as specified in Table 7-5~~Table 7-5~~. adaptive_ref_pic_marking_mode_flag shall be equal to 1 when the number of pictures that are currently marked as "used for long-term reference" is equal to Max(max_num_ref_frames, 1).

Table 7-5~~7-5~~ – Interpretation of adaptive_ref_pic_marking_mode_flag

adaptive_ref_pic_marking_mode_flag	Reference picture marking mode specified
0	Sliding window reference picture marking mode: A marking mode providing a first-in first-out mechanism for short-term reference pictures.
1	Adaptive reference picture marking mode: A reference picture marking mode providing syntax elements to specify marking of reference pictures as "unused for reference" and to assign long-term frame indices.

memory_management_control_operation specifies a control operation to be applied to affect the reference picture marking. The memory_management_control_operation syntax element is followed by data necessary for the operation specified by the value of memory_management_control_operation. The values and control operations associated with memory_management_control_operation are specified in Table 7-6~~Table 7-6~~. The memory_management_control_operation syntax elements are processed by the decoding process in the order in which they appear in the slice header, and the semantics constraints expressed for each memory_management_control_operation apply at the specific position in that order at which that individual memory_management_control_operation is processed.

memory_management_control_operation shall not be equal to 1 in a slice header unless the specified reference picture is marked as "used for short-term reference" when the memory_management_control_operation is processed by the decoding process.

memory_management_control_operation shall not be equal to 2 in a slice header unless the specified long-term picture number refers to a reference picture that is marked as "used for long-term reference" when the memory_management_control_operation is processed by the decoding process.

memory_management_control_operation shall not be equal to 3 in a slice header unless the specified reference picture is marked as "used for short-term reference" when the memory_management_control_operation is processed by the decoding process.

memory_management_control_operation shall not be equal to 3 or 6 if the value of the variable MaxLongTermFrameIdx is equal to "no long-term picture indices" when the memory_management_control_operation is processed by the decoding process.

Not more than one memory_management_control_operation equal to 4 shall be present in a slice header.

Not more than one memory_management_control_operation equal to 5 shall be present in a slice header.

Not more than one memory_management_control_operation equal to 6 shall be present in a slice header.

memory_management_control_operation shall not be equal to 5 in a slice header unless no memory_management_control_operation in the range of 1 to 3 is present in the same decoded reference picture marking syntax structure.

A memory_management_control_operation equal to 5 shall not follow a memory_management_control_operation equal to 6 in the same slice header.

When a memory_management_control_operation equal to 6 is present, any memory_management_control_operation equal to 2, 3, or 4 that follows the memory_management_control_operation equal to 6 within the same slice header shall not specify the current picture to be marked as "unused for reference".

NOTE 2 – These constraints prohibit any combination of multiple memory_management_control_operation syntax elements that would specify the current picture to be marked as "unused for reference". However, some other combinations of memory_management_control_operation syntax elements are permitted that may affect the marking status of other reference pictures more than once in the same slice header. In particular, it is permitted for a memory_management_control_operation equal to 3 that specifies a long-term frame index to be assigned to a particular short-term reference picture to be followed in the same slice header by a memory_management_control_operation equal to 2, 3, 4 or 6 that specifies the same reference picture to subsequently be marked as "unused for reference".

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000337

Table 7-6~~7-6~~ – Memory management control operation (memory_management_control_operation) values

memory_management_control_operation	Memory Management Control Operation
0	End memory_management_control_operation syntax element loop
1	Mark a short-term reference picture as "unused for reference"
2	Mark a long-term reference picture as "unused for reference"
3	Mark a short-term reference picture as "used for long-term reference" and assign a long-term frame index to it
4	Specify the maximum long-term frame index and mark all long-term reference pictures having long-term frame indices greater than the maximum value as "unused for reference"
5	Mark all reference pictures as "unused for reference" and set the MaxLongTermFrameIdx variable to "no long-term frame indices"
6	Mark the current picture as "used for long-term reference" and assign a long-term frame index to it

difference_of_pic_nums_minus1 is used (with memory_management_control_operation equal to 3 or 1) to assign a long-term frame index to a short-term reference picture or to mark a short-term reference picture as "unused for reference". When the associated memory_management_control_operation is processed by the decoding process, the resulting picture number derived from difference_of_pic_nums_minus1 shall be a picture number assigned to one of the reference pictures marked as "used for reference" and not previously assigned to a long-term frame index.

The resulting picture number shall be one of the set of picture numbers assigned to reference pictures.

long_term_pic_num is used (with memory_management_control_operation equal to 2) to mark a long-term reference picture as "unused for reference". When the associated memory_management_control_operation is processed by the decoding process, long_term_pic_num shall be equal to a long-term picture number assigned to one of the reference pictures that is currently marked as "used for long-term reference".

The resulting long-term picture number shall be one of the set of long-term picture numbers assigned to reference pictures.

long_term_frame_idx is used (with memory_management_control_operation equal to 3 or 6) to assign a long-term picture index to a picture. When the associated memory_management_control_operation is processed by the decoding process, the value of long_term_frame_idx shall be in the range of 0 to MaxLongTermFrameIdx, inclusive.

max_long_term_frame_idx_plus1 minus 1 specifies the maximum value of long-term picture index allowed for long-term reference pictures (until receipt of another value of max_long_term_frame_idx_plus1). The value of max_long_term_frame_idx_plus1 shall be in the range of 0 to max_num_ref_frames, inclusive.

7.4.3.4 Sample adaptive offset parameter semantics

sample_adaptive_offset_flag specifies whether sample adaptive offset applies or not for the current slice.

sao_split_flag[saoDepth][x0][y0] specifies whethere a region is split into four sub regions with half horizontal and vertical number of LCU. When sao_split_flag[saoDepth][x0][y0] is not present, it shall be inferred to be equal to 0.

The maximum allowed depth for sample adaptive offset process SaoMaxDepth is derived as follows:

SaoMaxDepth = Min(4, Min(Floor(Log2(PicWidthInLCUs)), Floor(Log2(PicHeightInLCUs)))) (7-~~13~~9)

[Ed.: (WJ) PicWidthInLCUs and PicHeightInLCUs should be defined elsewhere. Note: SAO assumes that PicWidthInLCUs = Round($PicWidthInSamples_L$ ÷ (1 << Log2MaxCUSize))

sao_type_idx[saoDepth][x0][y0] indicates the offset type for the region specified by saoDepth, x0 and y0.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000338

sao_offset[saoDepth][x0][y0][i] indicates the offset value of i-th category for the region specified by saoDepth, x0 and y0.

[Ed.: (WJ) x0 and y0 are not sample position here, but indices from 0 to 3 for 4-split sub region. Maybe better to use sample position for the consistency to other parts of the text]

The number of categories, NumSaoClass, is specified in Table 7-7.

Table 7-73~~7~~ – Specification of NumSaoClass

sao_type_idx[saoDepth][x0][y0]	NumSaoCategory	Edge type (informative)
0	0	Not applied
1	4	1D 0-degree edge
2	4	1D 90-degree edge
3	4	1D 135-degree edge
4	4	1D 45-degree edge
5	16	Central band
6	16	Side band

7.4.3.5 Adaptive loop filter parameter semantics

adaptive_loop_filter_flag specifies whether adaptive loop filter applies or not for the current slice.

alf_region_adaptation_flag specifies the filter adaptation method shall be applied for the current slice. If alf_region_adaptation_flag is equal to 1, the region-based filter adaptation shall be applied, otherwise, the block-based directional-activity filter adaptation shall be applied.

alf_length_luma_minus5_div2 specifies the filter length in horizontal direction for the luma component alfTap used in the adaptive loop filter process as specified by

$$alfTap = (alf\_length\_luma\_minus5\_div2 << 1) + 5 \quad (7\text{-}14\text{~~1~~}1)$$

The number of coded luma filter coefficients AlfCodedLengthLuma and the number of luma filter coefficients AlfLengthLuma are derived as follows:

– If alfTap is equal to 9, [Ed.: (WJ) truncated diamond shape]

$$AlfCodedLengthLuma = ((alfTap * alfTap) >> 2) + 1 \quad (7\text{-}15\text{~~1~~}1)$$

$$AlfLengthLuma = ((alfTap * alfTap) >> 1) \quad (7\text{-}16\text{~~1~~}2)$$

– Otherwise (alfTap is less than 9), [Ed.: (WJ) diamond shape]

$$AlfCodedLengthLuma = ((alfTap * alfTap) >> 2) + 2 \quad (7\text{-}17\text{~~1~~}3)$$

$$AlfLengthLuma = ((alfTap * alfTap) >> 1) + 2 \quad (7\text{-}18\text{~~1~~}4)$$

alf_no_filters_minus1 plus 1 specifies the number of filter sets for the current slice.

alf_start_second_filter specifies the variance index of luma samples where the second filter is applied, when alf_no_filters_minus1 is equal to 1.

alf_filter_pattern[i] specifies the filter index array corresponding to i-th variance index of luma samples, when alf_no_filters_minus1 is greater than 1. The number of filter sets AlfNumFilters is derived as follows:

– If alf_no_filters_minus1 is less than 2,

$$AlfNumFilters = alf\_no\_filters\_minus1 + 1 \quad (7\text{-}19\text{~~1~~}5)$$

– Otherwise (alf_no_filters_minus1 is greater than 2)

$$AlfNumFilters = \Sigma_i\ alf\_filter\_pattern[i] \text{ with } i = 0..15 \quad (7\text{-}20\text{~~1~~}6)$$

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000339

alf_pred_method specifies whether the filter coefficients are coded in a predictive way.

alf_min_kstart_minus1 plus 1 specifies the minimum order k of k-th order exponential golomb code for the luma filter coefficients for the adaptive loop filter.

alf_golomb_index_bit specifies the difference in order k of k-th order exponential golomb code for the different groups of the luma filter coefficients. Note that there are several groups of the luma filter coefficients where each group may have different order k.

alf_coeff_luma[i][j] specifies the j-th filter coefficient of i-th filter for the adaptive loop filtering process for luma samples.

alf_chroma_idc specifies which chroma components are to be filtered.

Table 7-87-83-8 – Specification of alf_chroma_idc

alf_chroma_idc	chroma component to be filtered
0	None
1	Cr
2	Cb
3	Cb and Cr

alf_length_chroma_minus5_div2 specifies the filter length in horizontal direction for the luma component alfTapC used in the adaptive loop filter process as specified by

alfTapC = (alf_length_chroma_minus5_div2 << 1) + 5 (7-21~~17~~)

The number of coded chroma filter coefficients AlfCodedLengthChroma and the number of chroma filter coefficients AlfLengthChroma are derived as follows:

AlfCodedLengthChroma = ((alfTapC * alfTapC – 1) >> 1) + 2 (7-22~~18~~)

AlfLengthChroma = (alfTapC * alfTapC) + 1 (7-23~~19~~)

alf_coeff_chroma[i] specifies the i-th filter coefficient for the adaptive loop filtering process for chroma samples.

7.4.3.6 Adaptive loop filter coding unit control parameter semantics

alf_cu_control_flag specifies whether the adaptive loop filter process for luma component shall be applied adaptively according to the coding unit. If alf_cu_control_flag is equal to 1, the filtering process shall be applied only when alf_cu_flag is equal to 1, otherwise, the filtering process shall be applied to all luma samples in the current slice.

alf_cu_control_max_depth specifies the maximum split depth from treeblock for transmitting the flag indicating the application of the adaptive loop filter process.

alf_length_cu_control_info specifies the information of the number of alf_cu_flag succeeding this information. ~~alf_length_cu_control_info shall present unless alf_cu_control_max_depth is 0, otherwise, it shall be inferred to be equal to 0.~~ When ~~The~~ the variable specifying the number of alf_cu_flag shall be computed as

NumAlfCuFlag = alf_length_cu_control_info + NumTBsInPicture (7-24~~20~~)

~~[Ed.: (WJ) NumTBsInPicture should be dependent on multiple slice structure]~~

alf_cu_flag[i] specifies the information whether a coding unit is filtered when the adaptive loop filter is applied.

7.4.4 Slice data semantics

end_of_slice_flag equal to 0 specifies that another macroblock is following in the slice. end_of_slice_flag equal to 1 specifies the end of the slice and that no further macroblock follows.

7.4.5 Coding tree semantics

split_coding_unit_flag[x0][y0] specifies whether a coding unit is split into coding units with half horizontal and vertical size. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000340

When split_coding_unit_flag[x0][y0] is not present, the following applies:

– If log2CUSize is greater than Log2MinCUSize, the value of split_coding_unit_flag[x0][y0] is inferred to be equal to 1.

– Otherwise (log2CUSize is equal to Log2MinCUSize), the value of split_coding_unit_flag[x0][y0] is inferred to be equal to 0.

cu_split_pred_part_mode[x0][y0] specifies split_coding_unit_flag and when the coding unit is not split the skip_flag[x0][y0], PredMode and PartMode of a coding unit. The array indices x0 and y0 specify the location (x0, y0) of the top-left luma sample of the coding unit relative to the top-left luma sample of the picture.

Table 7-83 – Specification of cu_split_pred_part_mode when log2CUSize is greater than Log2MinCUSize

cu_split_pred_part_mode	split_coding_unit_flag	merge_flag	PredMode	PartMode
0	1	-	-	-
1	0	-	MODE_SKIP	PART_2Nx2N
2	0	1	MODE_INTER	PART_2Nx2N
3	0	0	MODE_INTER	PART_2Nx2N
4	0	-	MODE_INTER	PART_2NxN
5	0	-	MODE_INTER	PART_Nx2N
6	0	-	MODE_INTRA	PART_2Nx2N

Table 7-107 – Specification of cu_split_pred_part_mode when log2CUSize is equal to Log2MinCUSize

cu_split_pred_part_mode	split_coding_unit_flag	merge_flag	PredMode	PartMode
0	0	-	MODE_SKIP	PART_2Nx2N
1	0	1	MODE_INTER	PART_2Nx2N
2	0	0	MODE_INTER	PART_2Nx2N
3	0	-	MODE_INTER	PART_2NxN
4	0	-	MODE_INTER	PART_Nx2N
5 (escape symbol)	0	-	MODE_INTRA	PART_2Nx2N
			MODE_INTRA	PART_NxN
			MODE_INTER	PARN_NXN

7.4.6 Coding unit semantics

skip_flag[x0][y0] equal to 1 specifies that for the current coding unit, when decoding a P or B slice, no more syntax elements except the motion vector predictor indices are parsed after skip_flag[x0][y0]. skip_flag[x0][y0] equal to 0 specifies that the coding unit is not skipped. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture.

When skip_flag[x0][y0] is not present, it shall be inferred to be equal to 0.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000341

intra_part_mode specifies whether the PartMode for the current coding unit is PART_2Nx2N or PART_NxN, when the slice type is I and log2CUSize is equal to Log2MinCUSize. If intra_part_mode is equal to 1, the PartMode is PART_2Nx2N. Otherwise if intra_part_mode is equal to 0, the PartMode is PART_NxN.

pred_type specifies prediction mode and partitioning type of the current coding unit. The semantics of pred_type depend on the slice type. The variables PredMode, PartMode and IntraSplitFlag are derived from the value of pred_type as defined in Table 7-11~~Table 7-8~~. The value of pred_type shall not be equal to 3 or 5 when slice_type is equal to P or B and log2CUSize is greater than Log2MinCUSize.

When pred_type is not present, the variables PredMode and PartMode are derived as follows.

– If slice_type is equal to I and log2CUSize is greater than Log2MinCUSize,

  -PredMode is inferred to be equal to MODE_INTRA

  -PartMode is inferred to be equal to PART_2Nx2N

  -IntraSplitFlag is inferred to be equal to 0

– Otherwise (slice_type is equal to P or B), if skip_flag[x0][y0] is equal to 1,

  -PredMode is inferred to be equal to MODE_SKIP

  -PartMode is inferred to be equal to PART_2Nx2N

pcm_flag specifies whether the associated coding unit with PART_2Nx2N is coded by I_PCM. If the pcm_flag is equal to 1, the associated coding unit with PART_2Nx2N is coded by I_PCM. When the pcm_flag is not present, it shall be inferred to be equal to 0.

pcm_alignment_zero_bit is a bit equal to 0.

pcm_sample_luma[i] represents a coded luma sample value in the raster scan within the coding unit. The number of bits used to represent each of these samples is PCMBitDepth$_Y$.

pcm_sample_chroma[i] represents a coded chroma sample value in the raster scan within the coding unit. The first half of the values represent coded Cb samples and the remaining half of the values represent coded Cr samples. The number of bits used to represent each of these samples is PCMBitDepth$_C$.

Table 7-11~~7-11~~ - Name association to prediction mode and partitioning type

slice_type	pred_type	IntraSplitFlag	PredMode	PartMode
I	0	0	MODE_INTRA	PART_2Nx2N
	1	1	MODE_INTRA	PART_NxN
P or B	0	-	MODE_INTER	PART_2Nx2N
	1	-	MODE_INTER	PART_2NxN
	2	-	MODE_INTER	PART_Nx2N
	3	-	MODE_INTER	PART_NxN
	4	0	MODE_INTRA	PART_2Nx2N
	5	1	MODE_INTRA	PART_NxN
	inferred	-	MODE_SKIP	PART_2Nx2N

7.4.7 Prediction unit semantics

mvp_idx_l0[x0][y0] specifies the motion vector predictor index of list 0 where x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered prediction block relative to the top-left luma sample of the picture.

When mvp_idx_l0[x0][y0] is not present, it shall be inferred to be equal to 0.

mvp_idx_l1[x0][y0] has the same semantics as mvp_idx_l0, with l0 and list 0 replaced by l1 and list 1, respectively.

mvp_idx_lc[x0][y0] has the same semantics as mvp_idx_l0, with l0 and list 0 replaced by lc and list combination, respectively.

prev_intra_luma_pred_flag[x0][y0], **mpm_idx[** x0 **][** y0 **]** and **rem_intra_luma_pred_mode**[x0][y0] specify the intra prediction mode for luma samples. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered prediction block relative to the top-left luma sample of the picture. When

서식 있음: 글꼴: 굵게

서식 있음: 글꼴: 굵게

서식 있음: 글꼴: 굵게

서식 있음: 아래 첨자

서식 있음: 글꼴: 굵게

서식 있음: 글머리 기호 또는 번호 없이, 탭: 1.42 글자(없음)

서식 있음: 아래 첨자

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000342

prev_intra_luma_pred_flag[x0][y0] is equal to 1, the intra prediction mode is inferred from a neighbouring intra-predicted prediction unit according to subclause 8.3.1~~8.3.1~~.

When **mpm_idx[** x0 **][** y0 **]** is not present, it shall be inferred to have a value of 0.

The variable IntraLumaModeBins specifies the number of bins indicating the rem_intra_luma_pred_mode[x0][y0]. IntraLumaModeBins depends on the size of the prediction unit and limits the number of available prediction modes as given by Table 7-12~~Table 7-11~~.

Table 7-~~12~~~~7-11~~ – Number of bins and number of mode for rem_intra_luma_pred_mode associated to PuSize

PuSize	IntraLumaModeBins	Number of Modes
~~PU_2x2~~	~~1~~	~~3~~
PU_4x4	4	17
PU_8x8	5	34
PU_16x16	5	34
PU_32x32	5	34
PU_64x64	1	3
~~PU_128x128~~	~~2~~	~~5~~

planar_flag_luma[x0][y0] specifies whether the luma block shall be derived by the planar prediction when IntraPredModeY[x0][y0] is equal to 2. If planar_flag_luma[x0][y0] is not present, planar_flag_luma[x0][y0] shall be inferred to be equal to 0.

intra_chroma_pred_mode[x0][y0] specifies the intra prediction mode for chromas samples. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered prediction block relative to the top-left luma sample of the picture.

planar_flag_chroma[x0][y0] specifies whether the chroma block shall be derived by the planar prediction when IntraPredModeC is equal to 2.

If planar_flag_chroma[x0][y0] is not present (SignaledAsChromaDC is not equal to 1 or IntraPredModeC[x0][y0] is equal to Intra_DC or IntraPredModeC[x0][y0] is equal to Intra_Planar), planar_flag_chroma[x0][y0] shall be inferred as follows:

- If SignaledAsChromaDC is not equal to 1, planar_flag_chroma[x0][y0] shall be inferred to be equal to 0.
- Otherwise, if IntraPredModeC[x0][y0] is equal to Intra_DC or Intra_Planar, the following applies.
  - If chroma_pred_from_luma_enabled_flag is equal to 1, the following applies.
    - If intra_chroma_pred_mode[x0][y0] is equal to 3, planar_flag_chroma[x0][y0] is set equal to planar_flag_luma[x0][y0]. [Ed. (WJ): possible conflict with codeword switch]
    - Otherwise (intra_chroma_pred_mode[x0][y0] is equal to 4), planar_flag_chroma[x0][y0] is set equal to planar_flag_luma[x0][y0].
  - Otherwise (chroma_pred_from_luma_enabled_flag is equal to 0), the following applies.
    - If intra_chroma_pred_mode[x0][y0] is equal to 2, planar_flag_chroma[x0][y0] is set equal to 1 planar_flag_luma[x0][y0]. [Ed. (WJ): possible conflict with codeword switch]
    - Otherwise (intra_chroma_pred_mode[x0][y0] is equal to 3), planar_flag_chroma[x0][y0] is set equal to planar_flag_luma[x0][y0].

[Ed. (WJ): suggested to be noted from proponents – under the specific value of intra_chroma_pred_mode, the planar_flag_chroma may be inferred based on planar_flag_luma. The binarization of intra_chroma_pred_mode should take account of this property and the codeword switching as listed in the open issue]

.....

서식 있음 ... [1]
서식 있음 ... [2]
서식 있음 ... [3]
서식 있음 ... [4]
서식 있음 ... [5]
서식 있음 ... [6]
서식 있음 ... [7]
서식 있음 ... [8]
서식 있음 ... [9]
서식 있음 ... [10]
서식 있음 ... [11]
서식 있음 ... [12]
서식 있음 ... [13]
서식 있음 ... [14]
서식 있음 ... [15]
서식 있음 ... [16]
서식 있음 ... [17]
서식 있음 ... [18]
서식 있음 ... [19]
서식 있음 ... [20]
서식 있음 ... [21]
서식 있음 ... [22]
서식 있음 ... [23]
서식 있음 ... [24]
서식 있음 ... [25]
서식 있음 ... [26]
서식 있음 ... [27]
서식 있음 ... [28]
서식 있음 ... [29]
서식 있음 ... [30]
서식 있음 ... [31]
서식 있음 ... [32]
서식 있음 ... [33]
서식 있음 ... [34]
서식 있음 ... [35]
서식 있음 ... [36]
서식 있음 ... [37]
서식 있음 ... [38]
서식 있음 ... [39]
서식 있음 ... [40]
서식 있음 ... [41]
서식 있음 ... [42]
서식 있음 ... [43]
서식 있음 ... [44]
서식 있음 ... [45]
서식 있음 ... [46]

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000343

merge_flag[x0][y0] specifies whether the inter prediction parameters for the current prediction unit are inferred from a neighbouring inter-predicted partition. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered prediction block relative to the top-left luma sample of the picture.

When merge_flag[x0][y0] is not present (InferredMergeFlag is equal to 1), it is inferred to be equal to 1.

merge_idx[x0][y0] specifies the merging candidate index of the merging candidate list where x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered prediction block relative to the top-left luma sample of the picture.

When merge_idx[x0][y0] is not present, it is inferred to be equal to 0.

combined_inter_pred_ref_idx specifies the values of inter_pred_flag and ref_idx_lX with X being 0, 1 or c. When it is not present, it shall be inferred to be equal to 0. [Ed. (WJ): needs to be improved]

The variables NumPredRefLC, NumPredRefL0 and NumPredRefL1 are specified as

NumPredRefLC = Min(4, num_ref_idx_lc_active_minus1 + 1) (7-25~~18~~)
NumPredRefL0 = Min(2, num_ref_idx_l0_active_minus1 + 1) (7-26~~18~~)
NumPredRefL1 = Min(2, num_ref_idx_l1_active_minus1 + 1) (7-27~~18~~)

The variable MaxPredRef is specified as

– If num_ref_idx_lc_active_minus1 is greater than 0, the following applies.

MaxPredRef = NumPredRefLC + NumPredRefL0 * NumPredRefL1 (7-28~~18~~)

– Otherwise (combined list is not used), the following applies.

MaxPredRef = NumPredRefL0 + NumPredRefL0 * NumPredRefL1 (7-29~~18~~)

The function use_combined_inter_pred_ref(x, y) is specified as follows: [Ed. (WJ): move to syntax function section?]

```
use_combined_inter_pred_ref( x, y ) = check_inter_pred_ref_neighbour( x-1, y ) &&
                                      check_inter_pred_ref_neighbour( x, y-1 ) &&
                                      check_inter_pred_ref_neighbour( x-1, y-1 )
```

The function check_inter_pred_ref_neighbour(x, y) is specified as follows: [Ed. (WJ): move to syntax function section?]

```
if x or y is outside current slice, check_inter_pred_ref_neighbour( x, y ) = true
else if ( num_ref_idx_lc_active_minus1 <= 0 && ref_idx_l0[ x ][ y ] < 2 && ref_idx_l1[ x ][ y ] < 2 )
        check_inter_pred_ref_neighbour( x, y ) = true
else if ( num_ref_idx_lc_active_minus1 > 0 && inter_pred_flag[ x ][ y ] != PredLC &&
          ref_idx_l0[ x ][ y ] < 2 && ref_idx_l1[ x ][ y ] < 2 )
        check_inter_pred_ref_neighbour( x, y ) = true
else if ( num_ref_idx_lc_active_minus1 > 0 && inter_pred_flag[ x ][ y ] == PredLC && ref_idx_lc[ x ][ y ] < 4 )
        check_inter_pred_ref_neighbour( x, y ) = true
else
        check_inter_pred_ref_neighbour( x, y ) = false
```

inter_pred_flag[x0][y0] specifies whether uni-prediction, or bi-prediction is used for the current prediction unit according to Table 7-13~~Table 7-11~~. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered prediction block relative to the top-left luma sample of the picture.

Table 7-13~~7-11~~ – Name association to inter prediction mode

slice_type	inter_pred_flag	Name of inter_pred_flag
P	inferred	Pred_L0

서식 있음: 글꼴: 굵게
서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 1.42 글자, 왼쪽 + 4.26 글자(없음) + 5.67 글자 + 7.09 글자
서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자
서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 2.13 글자, 왼쪽 + 4.26 글자(없음) + 5.67 글자 + 7.09 글자
서식 있음: 영어(영국)
서식 있음: 영어(영국)
서식 있음: 영어(영국)
서식 있음: 영어(영국)
서식 있음: 영어(영국)
서식 있음: 영어(영국)
서식 있음: 영어(영국)
서식 있음: 영어(영국)
서식 있음: 영어(영국)
서식 있음: 탭: 2.13 글자, 왼쪽
서식 있음: 영어(영국)
서식 있음: 영어(영국)
서식 있음: 영어(영국)
서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자
서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 2.13 글자, 왼쪽 + 4.26 글자(없음) + 5.67 글자 + 7.09 글자

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000344

B	0	Pred_LC
	1	Pred_BI

When entropy_coding_mode_flag is equal to 1 and inter_pred_flag[x0][y0] is not present, it shall be inferred to be equal to Pred_L0 when slice_type is equal to P and Pred_BI when slice_type is equal to B.

When entropy_coding_mode_flag is equal to 0, inter_pred_flag[x0][y0] is not present and combined_inter_pred_ref_idx is present, it shall be inferred as follows:

- If num_ref_idx_lc_active_minus1 is greater than 0 and combined_inter_pred_ref_idx < NumPredRefLC, inter_pred_flag[x0][y0] is set equal to Pred_LC.
- Otherwise, if num_ref_idx_lc_active_minus1 is equal to 0 and combined_inter_pred_ref_idx < NumPredRefL0, inter_pred_flag[x0][y0] is set equal to Pred_L0.
- Otherwise, inter_pred_flag[x0][y0] is set equal to Pred_BI.

ref_idx_l0[x0][y0] specifies the list 0 reference picture index for the current prediction unit. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered prediction block relative to the top-left luma sample of the picture.

When ref_idx_l0[x0][y0] is not present and combined_inter_pred_ref_idx is present, ref_idx_l0[x0][y0] shall be inferred to be equal to ((combined_inter_pred_ref_idx – NumPredRefLC) / NumPredRefL1).

ref_idx_l0_minusX[x0][y0] + X specifies the value of ref_idx_l0[x0][y0] when combined_inter_pred_ref_idx is present and it is equal to MaxPredRef.

The constant X is derived as follows:

- If inter_pred_flag[x0][y0] is Pred_L0, X is set equal to 2.
- Otherwise, X is set equal to 0.

ref_idx_l1[x0][y0] has the same semantics as ref_idx_l0, with l0 and list 0 replaced by l1 and list 1, respectively.

When ref_idx_l1[x0][y0] is not present and combined_inter_pred_ref_idx is present, ref_idx_l1[x0][y0] shall be inferred to be equal to ((combined_inter_pred_ref_idx – NumPredRefLC) % NumPredRefL1).

ref_idx_l1_minusX[x0][y0] + X specifies the value of ref_idx_l1[x0][y0] when combined_inter_pred_ref_idx is present and it is equal to MaxPredRef.

The constant X is derived as follows:

- If ref_idx_l0[x0][y0] is less than 2, X is set equal to 2.
- Otherwise, X is set equal to 0.

ref_idx_lc[x0][y0] has the same semantics as ref_idx_l0, with l0 and list 0 replaced by lc and list combination, respectively.

When ref_idx_lc[x0][y0] is not present and combined_inter_pred_ref_idx is present, ref_idx_lc[x0][y0] shall be inferred to be equal to combined_inter_pred_ref_idx.

ref_idx_lc_minus4[x0][y0] + 4 specifies the value of ref_idx_lc [x0][y0] when combined_inter_pred_ref_idx is present and it is equal to MaxPredRef.

mvd_l0[x0][y0][compIdx], specifies the difference between a list 0 vector component to be used and its prediction. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered prediction block relative to the top-left luma sample of the picture. The horizontal motion vector component difference is assigned compIdx = 0 and the vertical motion vector component is assigned compIdx = 1.

When any of the two components is not present, the inferred value is 0.

mvd_l1[x0][y0][compIdx] has the same semantics as mvd_l0, with l0 and list 0 replaced by l1 and list 1, respectively.

서식 있음: 글머리 기호 + 수준:1 + 맞춤 위치: 0 cm + 탭 간격: 0.69 cm + 들여쓰기 위치: 0.69 cm

서식 있음: 글꼴: (한글) 맑은 고딕

서식 있음: 글머리 기호 + 수준:1 + 맞춤 위치: 0 cm + 탭 간격: 0.69 cm + 들여쓰기 위치: 0.69 cm

서식 있음: 글머리 기호 + 수준:1 + 맞춤 위치: 0 cm + 탭 간격: 0.69 cm + 들여쓰기 위치: 0.69 cm

서식 있음: 영어(영국)

서식 있음: 글꼴: (한글) 맑은 고딕, 영어(영국)

서식 있음: 글꼴: 굵게

서식 있음: 영어(미국)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000345

mvd_lc[x0][y0]**[** compIdx **]** has the same semantics as mvd_l0, with l0 and list 0 replaced by lc and list combination, respectively.

7.4.8 Transform tree semantics

no_residual_data_flag equal to 1 specifies that no residual data are present for the current coding unit. no_residual_data_flag equal to 0 specifies that residual data are present for the current coding unit.

When no_residual_data_flag is not present, its value shall be inferred to be equal to 0.

split_transform_flag[x0][y0][trafoDepth] specifies whether a block is split into blocks with half horizontal and vertical size for the purpose of transform coding. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered block relative to the top-left luma sample of the picture. The array index trafoDepth specifies the current subdivision level of a coding unit into blocks for the purpose of transform coding. trafoDepth is equal to 0 for blocks that correspond to coding units.

When split_transform_flag[x0][y0][trafoDepth] is not present, it is inferred as follows:

– If log2TrafoSize is greater than Log2MaxTrafoSize or intraSplitFlag is equal to 1, the value of split_transform_flag[x0][y0][trafoDepth] is inferred to be equal to 1.

– Otherwise (log2TrafoSize is less than or equal to Log2MaxTrafoSize and intraSplitFlag is equal to 0), the value of split_transform_flag[x0][y0][trafoDepth] is inferred to be equal to 0.

cbf_luma[x0][y0][trafoDepth] equal to 1 specifies that the luma transform block contains one or more transform coefficient levels not equal to 0. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture. The array index trafoDepth specifies the current subdivision level of a coding unit into blocks for the purpose of transform coding. trafoDepth is equal to 0 for blocks that correspond to coding units.

When cbf_luma[x0][y0][trafoDepth] is not present, it is inferred to be equal to 1.

cbf_cb[x0][y0][trafoDepth] equal to 1 specifies that the Cb transform block contains one or more transform coefficient levels not equal to 0. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture. The array index trafoDepth specifies the current subdivision level of a coding unit into blocks for the purpose of transform coding. trafoDepth is equal to 0 for blocks that correspond to coding units.

When cbf_cb[x0][y0][trafoDepth] is not present and PredMode is not equal to MODE_INTRA, the value of cbf_cb[x0][y0][trafoDepth] is inferred to be equal to 0.

cbf_cr[x0][y0][trafoDepth] equal to 1 specifies that the Cr transform block contains one or more transform coefficient levels not equal to 0. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture. The array index trafoDepth specifies the current subdivision level of a coding unit into blocks for the purpose of transform coding. trafoDepth is equal to 0 for blocks that correspond to coding units.

When cbf_cr[x0][y0][trafoDepth] is not present and PredMode is not equal to MODE_INTRA, the value of cbf_cr[x0][y0][trafoDepth] is inferred to be equal to 0.

cbp_and_split_transform contains a number of coded block flags and/or a split_transform_flag. Depending on a 4-bit value flagPattern defined as follows, the value of cbp_and_split_transform is defined according to Table 7-14~~Table 7-13~~.

```
codeLuma = trafoDepth == 0 || cbf_luma[ xB ][ yB ][ trafoDepth − 1 ]
codeCb =   trafoDepth == 0 || (cbf_cb[ xB ][ yB ][ trafoDepth − 1 ] &&
           log2TrafoSize > Log2MinTrafoSize )
codeCr =   trafoDepth == 0 || (cbf_cr[ xB ][ yB ][ trafoDepth − 1 ] &&
           log2TrafoSize > Log2MinTrafoSize )                                (7-30~~33~~)
codeSplitTrans = log2TrafoSize <= Log2MaxTrafoSize &&
                 log2TrafoSize > Log2MinTrafoSize &&
                 trafoDepth < maxDepth &&
                 !intraSplitFlag
flagPattern = (codeLuma << 3) + (codeCb << 2) + (codeCr << 1) + codeSplitTrans
```

The values of codeLuma, codeCb, codeCr and codeSplitTrans specify respectivley if an associated flag of cbf_luma, cbf_cb, cbf_cr, and split_transform_flag is included in the syntax element cbp_and_split_transform.

In Table 7-14~~Table 7-13~~, Luma, Cb, Cr and Split denote the values of cbf_luma[x0][y0][trafoDepth], cbf_cb[x0][y0][trafoDepth], cbf_cr[x0][y0][trafoDepth] and split_transform_flag[x0][y0][trafoDepth]

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000346

respectively. Luma1, Luma2 and Luma3 denote respectively the values of cbf_luma[x1][y0][trafoDepth], cbf_luma[x0][y1][trafoDepth] and cbf_luma[x1][y1][trafoDepth]. The values of x1 and y1 are derived as

$$x1 = x0 + ((1 << log2TrafoSize) >> 1) \quad (7\text{-}3124)$$

$$y1 = y0 + ((1 << log2TrafoSize) >> 1) \quad (7\text{-}3224)$$

Table 7-147-148-14 – Specification of cbp_and_split_transform

flagPattern	cbp_and_split_transform
8	(Luma<<3) + (Luma1<<2) + (Luma2<<1) + Luma3 - 1
9	(Luma<<1) + Split
10	(Luma<<1) + Cr
11	(Luma<<2) + (Cr<<1) + Split
12	(Luma<<1) + Cb
13	(Luma<<2) + (Cb<<1) + Split
14	(Luma<<2) + (Cb<<1) + Cr
15	(Luma<<2) + ((Cb|Cr)<<1) + Split

7.4.9 Transform coefficient semantics

The transform coefficient levels are parsed into the arrays transCoeffLevel[x0][y0][trafoDepth][cIdx][n]. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture. The array index trafoDepth specifies the current subdivision level of a coding unit into blocks for the purpose of transform coding. trafoDepth is equal to 0 for blocks that correspond to coding units. The array index cIdx specifies an indicator for the colour component; it is equal to 0 for luma, equal to 1 for Cb, and equal to 2 for Cr. The array index n specifies the scanning position of the transform coefficient levels.

cu_qp_delta can change the value of QP_Y for a quantization group of coding units where the quantization group of coding units is specified as follows:

- If a coding unit with the split_coding_unit_flag[x0][y0] equal to 0 and the log2CUSize is greater than or equal to log2MinCUDQPSize, the quantization group includes this coding unit only.
- Otherwise, if a coding unit with the split_coding_unit_flag[x0][y0] equal to 1 and the log2CUSize is equal to log2MinCUDQPSize, the quantization group includes all coding units split from this coding unit.

The decoded value of cu_qp_delta shall be in the range of $-(26+ QpBdOffset_Y / 2)$ to $+(25+ QpBdOffset_Y / 2)$, inclusive. cu_qp_delta shall be inferred to be equal to 0 when it is not present for any quantization group of coding units.

The value of QP_Y is derived as

$$QP_Y = (((QP_{Y,PREV} + cu\_qp\_delta +52+ 2*QpBdOffset_Y)\%(52 + QpBdOffset_Y)) - QpBdOffset_Y \quad (7\text{-}3324)$$

where $QP_{Y,PREV}$ is the luma quantization parameter, QP_Y, of the left neighbor quantization group of coding units in the current slice. If the left neighbor quantization group in the current slice is not available, $QP_{Y,PREV}$ is the luma quantization parameter, of the previous quantization group in decoding order in the current slice. For the first quantization group of coding units in the slice $QP_{Y,PREV}$ is initially set equal to $SliceQP_Y$ at the start of each slice.

The value of QP'_Y is derived as

$$QP'_Y = QP_Y + QpBdOffset_Y \quad (7\text{-}3433)$$

When PredMode is equal to MODE_INTRA, different scanning orders are used. The arrays specifying the scanning order are derived as follows.

- The array ScanType[log2TrafoSize-2][IntraPredMode], specifying the scanning order for various transform block sizes and luma intra prediction modes, is derived as specified in Table 7-15Table 7-13.

서식 있음: 들여쓰기: 왼쪽: 0 cm, 내어쓰기: 4.26 글자, 탭: 2.13 글자, 왼쪽 + 2 글자(없음)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000347

Table 7-15 – Specification of ScanType[log2TrafoSize][IntraPredMode]

IntraPredMode	log2TrafoSize			
	2	3	4	5
0	1	1	0	0
1	2	2	0	0
2-3	0	0	0	0
4-5	1	1	0	0
6	0	0	0	0
7-8	2	2	0	0
9-10	0	0	0	0
11-12	1	1	0	0
13-14	0	0	0	0
15-16	2	2	0	0
17-19	0	0	0	0
20-23	1	1	0	0
24-27	0	0	0	0
28-31	2	2	0	0
32-33	0	0	0	0
34	0	0	0	0
35	n/a0	n/a0	n/a0	n/a0

The array ScanTypeC[log2TrafoSize][IntraPredModeC], specifying the scanning order for various transform block sizes and chroma intra prediction modes, is set equal to derived as follows.specified in Table 7-13. The scanning order for IntraPredModeC equal to 4 is set equal to the corresponding luma scanning order ScanType[log2TrafoSize][IntraPredMode].

If IntraPredModeC is less than 4, ScanTypeC[log2TrafoSize][intraPredModeC] is set equal to ScanType[log2TrafoSize + 1][IntraPredModeC].

Otherwise (IntraPredModeC is equal to 4), ScanTypeC[log2TrafoSize][intraPredModeC] is set equal to ScanType[log2TrafoSize – 1][intraPredMode].

[Ed. (WJ): the index log2TrafoSize + 1 is valid only for 4:2:0 chroma sampling]

Table 7-16 – Specification of ScanTypeC[log2TrafoSize][IntraPredModeC]

IntraPredModeC	log2TrafoSize			
	2	3	4	5
0	1	0	0	0
1	2	0	0	0
2-3	0	0	0	0
4	ScanType[log2TrafoSize][IntraPredMode]			

7.4.10 Residual coding CABAC semantics

The array ScanOrder[log2BlockSize – 2][scanIdx][pos][comp] specifies the mapping of the scan position pos, ranging from 0 to ((1 << log2BlockSize) * (1 << log2BlockSize)) – 1, to horizontal and vertical components of the scan-order matrix. The array index scanIdx equal to 0 specifies a zig-zag scan, scanIdx equal to 1 specifies a horizontal scan and scanIdx equal to 2 speicifes a vertical scan. The array index comp equal to 0 specifies the horizontal component and the array index comp equal to 1 specifies the vertical component. The array ScanOrder is derived as follows.

서식 있음: 들여쓰기: 왼쪽: 0 cm, 내어쓰기: 4.26 글자, 탭: 2.13 글자, 왼쪽 + 2 글자(없음)

서식 있음: 글머리 기호 또는 번호 없이, 탭: 4.26 글자, 왼쪽 + 2 글자(없음)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000348

The scanning order array initialisation process as specified in 6.5 is invoked with 1 << log2TrafoSize − 2 as input and the output is assigned to ScanOrder[log2TrafoSize − 2].

last_significant_coeff_x, last_significant_coeff_y specify the column and the row position of the last significant coefficient in scanning order within a transform block. The values of last_significant_coeff_x and last_significant_coeff_y are ranging from 0 to (1 << log2TrafoSize) − 1.

When scanIdx is equal to 2, the coordinates are swapped as follows.

```
temp                   = last_significant_coeff_x
last_significant_coeff_x = last_significant_coeff_y                    (7-352-3)
last_significant_coeff_y = temp
```

significant_coeff_flag[xC **][** yC **]** specifies for the transform coefficient position (xC, yC) within the current transform block whether the corresponding transform coefficient level at location (xC, yC) is non-zero as follows.

– If significant_coeff_flag[xC][yC] is equal to 0, the transform coefficient level at location (xC, yC) is set equal to 0;

– Otherwise (significant_coeff_flag[xC][yC] is equal to 1), the transform coefficient level at location (xC, yC) has a non-zero value.

The significant_coeff_flag[last_significant_coeff_x][last_significant_coeff_y] at the last significant location (last_significant_coeff_x, last_significant_coeff_y) in scan order is inferred to be equal to 1.

When significant_coeff_flag[xC][yC] is not present, it is inferred to be equal to 0.

coeff_abs_level_greater1_flag[n **]** specifies for the scanning position n whether there are transform coefficient levels greater than 1.

When coeff_abs_level_greater1_flag[n] is not present, it is inferred to be equal to 0.

coeff_abs_level_greater2_flag[n **]** specifies for the scanning position n whether there are transform coefficient levels greater than 2.

When coeff_abs_level_greater2_flag[n] is not present, it is inferred to be equal to 0.

coeff_abs_level_minus3[n **]** is the absolute value of a transform coefficient level minus 3 at the scanning position n. The value of coeff_abs_level_minus3 is constrained by the limits in subclause XX.

When coeff_abs_level_minus3[n] is not present, it is inferred as follows.

– If coeff_abs_level_greater1_flag[n] is equal to 0, coeff_abs_level_minus3[n] is inferred to be equal to −2.

– Otherwise (coeff_abs_level_greater1_flag[n] is equal to 1), coeff_abs_level_minus3[n] is inferred to be equal to −1.

coeff_sign_flag[n **]** specifies the sign of a transform coefficient level for the scanning position n as follows.

– If coeff_sign_flag[n] is equal to 0, the corresponding transform coefficient level has a positive value.

– Otherwise (coeff_sign_flag[n] is equal to 1), the corresponding transform coefficient level has a negative value.

When coeff_sign_flag[n] is not present, it is inferred to be equal to 0.

7.4.11 Residual coding CAVLC semantics

The following CAVLC syntax elements are used to decode a block of nMax transform coefficients.

last_pos_level_one specifies the position of the last non-zero valued transform coefficient in the scan and whether the absolute value of the transform coefficient is larger than 1.

– The position of the last non-zero transform coefficient in scan order is set equal to last_pos_level_one%nMax.

– If last_pos_level_one is less than nMax, the absolute value of the transform coefficient is equal to 1.

– Otherwise (last_pos_level_one is not less than nMax), the absolute value of the transform coeficient is larger than 1.

level_minus2_and_sign specifies the value of a transform coefficient that has an absolute value that is larger than one as follows.

– The absolute value of the transform coefficient is equal to (level_minus2_and_sign>>1) + 2.

– If (level_minus2_and_sign & 1) is equal to 0, the transform coefficient has a positive value.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000349

– Otherwise ((level_minus2_and_sign & 1) is equal to 1), the transform coefficient has a negative value.

sign_flag specifies the sign of a non-zero valued transform coefficient as follows.

– If sign_flag is equal to 0, the transform coefficient has a positive value.

– Otherwise (sign_flag is equal to 1), the transform coefficient has a negative value.

run_level_one specifies the number of consecutive transform coefficients in the scan with zero value before a non-zero valued transform coefficients and whether the absolute value of the non-zero valued transform coefficient is larger than 1.

– Given the scan position n of the current non-zero valued transform coefficient, the number of consecutive transform coefficients with zero value before the next non-zero valued transform coeficient is set equal to run_level_one%n.

– If run_level_one is less than n, the absolute value of the transform coefficient is equal to 1.

– Otherwise (run_level_one is not less than n), the absolute value of the next non-zero valued transform coefficient is larger than 1.

level specifies the absolute value of a non-zero valued transform coefficient.

8 Decoding process

Outputs of this process are decoded samples of the current picture (sometimes referred to by the variable CurrPic).

Depending on the value of chroma_format_idc, the number of sample arrays of the current picture is as follows.

– If chroma_format_idc is equal to 0, the current picture consists of 1 sample array S_L.

– Otherwise (chroma_format_idc is not equal to 0), the current picture consists of 3 sample arrays S_L, S_{Cb}, S_{Cr}.

This clause describes the decoding process, given syntax elements and upper-case variables from clause 6.6.7.

The decoding process is specified 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.

Each picture referred to in this clause is a complete primary coded picture or part of a primary coded picture. Each slice referred to in this clause is a slice of a primary coded picture. Each slice data partition referred to in this clause is a slice data partition of a primary coded picture.

Depending on the value of separate_colour_plane_flag, the decoding process is structured as follows.

– If separate_colour_plane_flag is equal to 0, the decoding process is invoked a single time with the current picture being the output.

– Otherwise (separate_colour_plane_flag is equal to 1), the decoding process is invoked three times. Inputs to the decoding process are all NAL units of the primary coded picture with identical value of colour_plane_id. The decoding process of NAL units with a particular value of colour_plane_id is specified as if only a coded video sequence with monochrome colour format with that particular value of colour_plane_id would be present in the bitstream. The output of each of the three decoding processes is assigned to the 3 sample arrays of the current picture with the NAL units with colour_plane_id equal to 0 being assigned to S_L, the NAL units with colour_plane_id equal to 1 being assigned to S_{Cb}, and the NAL units with colour_plane_id equal to 2 being assigned to S_{Cr}.

NOTE – The variable ChromaArrayType is derived as 0 when separate_colour_plane_flag is equal to 1 and chroma_format_idc is equal to 3. In the decoding process, the value of this variable is evaluated resulting in operations identical to that of monochrome pictures with chroma_format_idc being equal to 0.

An overview of the decoding process is given as follows:

1. The decoding of NAL units is specified in subclause 8.1.
2. The processes in subclause 8.2 specify decoding processes using syntax elements in the slice layer and above:
   – Variables and functions relating to picture order count are derived in subclause 8.2.1. (only needed to be invoked for one slice of a picture)
   – [Ed. (TW): clarify the following paragraph] When the frame_num of the current picture is not equal to PrevRefFrameNum and is not equal to (PrevRefFrameNum + 1) % MaxFrameNum, the decoding process for gaps in frame_num is performed according to subclause 8.2.3.2 prior to the decoding of any slices of the current picture.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000350

– At the beginning of the decoding process for each P, SP, or B slice, the decoding process for reference picture lists construction specified in subclause 8.2.2 is invoked for derivation of reference picture list 0 (RefPicList0), and when decoding a B slice, reference picture list 1 (RefPicList1).

– When the current picture is a reference picture and after all slices of the current picture have been decoded, the decoded reference picture marking process in subclause 8.2.2.4 specifies how the current picture is used in the decoding process of inter prediction in later decoded pictures.

3. The processes in subclauses 8.3, ABC specify decoding processes using syntax elements in the treeblock layer and above.

[Ed. (TW): insert corresponding text]

8.1 NAL unit decoding process

Inputs to this process are NAL units.

Outputs of this process are the RBSP syntax structures encapsulated within the NAL units.

The decoding process for each NAL unit extracts the RBSP syntax structure from the NAL unit and then operates the decoding processes specified for the RBSP syntax structure in the NAL unit as follows.

Subclause 8.2 describes the decoding process for NAL units with nal_unit_type equal to 1 and 5.

NAL units with nal_unit_type equal to 7 and 8 contain sequence parameter sets and picture parameter sets, respectively. Picture parameter sets are used in the decoding processes of other NAL units as determined by reference to a picture parameter set within the slice headers of each picture. Sequence parameter sets are used in the decoding processes of other NAL units as determined by reference to a sequence parameter set within the picture parameter sets of each sequence.

8.2 Slice decoding process

8.2.1 Decoding process for picture order count

[Ed. (TW): clarify the following paragraphs]

8.2.1.1 Decoding process for picture order count type 0

8.2.1.2 Decoding process for picture order count type 1

8.2.1.3 Decoding process for picture order count type 2

8.2.2 Decoding process for reference picture lists construction

[Ed. (TW): clarify the following paragraphs]

This process is invoked at the beginning of the decoding process for each P, SP, or B slice.

Decoded reference pictures are marked as "used for short-term reference" or "used for long-term reference" as specified by the bitstream and specified in subclause 8.2.2.4. Short-term reference pictures are identified by the value of frame_num. Long-term reference pictures are assigned a long-term picture index as specified by the bitstream and specified in subclause 8.2.2.4.

Subclause 8.2.2.1 is invoked to specify

– the assignment of variables FrameNum, FrameNumWrap, and PicNum to each of the short-term reference pictures, and

– the assignment of variable LongTermPicNum to each of the long-term reference pictures.

Reference pictures are addressed through reference indices as specified in subclause 오류! 참조 원본을 찾을 수 없습니다. Error! Reference source not found.. A reference index is an index into a reference picture list. When decoding a P or SP slice, there is a single reference picture list RefPicList0. When decoding a B slice, there is a second independent reference picture list RefPicList1 in addition to RefPicList0.

At the beginning of the decoding process for each slice, reference picture list RefPicList0, and for B slices RefPicList1, are derived as specified by the following ordered steps:

1. An initial reference picture list RefPicList0 and for B slices RefPicList1 are derived as specified in subclause 8.2.2.2.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000351

2. When ref_pic_list_modification_flag_l0 is equal to 1 or, when decoding a B slice, ref_pic_list_modification_flag_l1 is equal to 1, the initial reference picture list RefPicList0 and, for B slices, RefPicList1 are modified as specified in subclause 8.2.2.3.

NOTE – The modification process for reference picture lists specified in subclause 8.2.2.3 allows the contents of RefPicList0 and for B slices RefPicList1 to be modified in a flexible fashion. In particular, it is possible for a picture that is currently marked "used for reference" to be inserted into RefPicList0 and for B slices RefPicList1 even when the picture is not in the initial reference picture list derived as specified in subclause 8.2.2.2.

The number of entries in the modified reference picture list RefPicList0 is num_ref_idx_l0_active_minus1 + 1, and for B slices the number of entries in the modified reference picture list RefPicList1 is num_ref_idx_l1_active_minus1 + 1. A reference picture may appear at more than one index in the modified reference picture lists RefPicList0 or RefPicList1.

8.2.2.1 Decoding process for picture numbers

This process is invoked when the decoding process for reference picture lists construction specified in subclause 8.2.2, the decoded reference picture marking process specified in subclause 8.2.2.4, or the decoding process for gaps in frame_num specified in subclause 8.2.3.2 is invoked.

The variables FrameNum, FrameNumWrap, PicNum, LongTermFrameIdx, and LongTermPicNum are used for the initialisation process for reference picture lists in subclause 8.2.2.2, the modification process for reference picture lists in subclause 8.2.2.3, the decoded reference picture marking process in subclause 8.2.2.4, and the decoding process for gaps in frame_num in subclause 8.2.3.2.

To each short-term reference picture the variables FrameNum and FrameNumWrap are assigned as follows. First, FrameNum is set equal to the syntax element frame_num that has been decoded in the slice header(s) of the corresponding short-term reference picture. Then the variable FrameNumWrap is derived as

```
if( FrameNum > frame_num )
    FrameNumWrap = FrameNum − MaxFrameNum                    (8-1)
else
    FrameNumWrap = FrameNum
```

where the value of frame_num used in Equation ~~8-1~~ 8-1 is the frame_num in the slice header(s) for the current picture.

Each long-term reference picture has an associated value of LongTermFrameIdx (that was assigned to it as specified in subclause 8.2.2.4).

To each short-term reference picture a variable PicNum is assigned, and to each long-term reference picture a variable LongTermPicNum is assigned.

$$PicNum = FrameNumWrap \tag{8-2}$$

$$LongTermPicNum = LongTermFrameIdx \tag{8-3}$$

8.2.2.2 Initialisation process for reference picture lists

This initialisation process is invoked when decoding a P or B slice header.

RefPicList0 and RefPicList1 have initial entries as specified in subclauses 8.2.2.2.1 through 8.2.2.2.2.

When the number of entries in the initial RefPicList0 or RefPicList1 produced as specified in subclauses 8.2.2.2.1 through 8.2.2.2.2 is greater than num_ref_idx_l0_active_minus1 + 1 or num_ref_idx_l1_active_minus1 + 1, respectively, the extra entries past position num_ref_idx_l0_active_minus1 or num_ref_idx_l1_active_minus1 are discarded from the initial reference picture list.

When the number of entries in the initial RefPicList0 or RefPicList1 produced as specified in subclauses 8.2.2.2.1 through 8.2.2.2.2 is less than num_ref_idx_l0_active_minus1 + 1 or num_ref_idx_l1_active_minus1 + 1, respectively, the remaining entries in the initial reference picture list are set equal to "no reference picture".

서식 있음: 제목 5

8.2.2.2.1 Initialisation process for the reference picture list for P slices

This initialisation process is invoked when decoding a P slice in a coded picture.

When this process is invoked, there shall be at least one reference picture that is currently marked as "used for reference" (i.e., as "used for short-term reference" or "used for long-term reference") and is not marked as "non-existing". Pictures with higher values of temporal_id than the current picture cannot be used for reference, and are not included in the reference picture list.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000352

The reference picture list RefPicList0 is ordered so that short-term reference pictures have lower indices than long-term reference pictures.

The short-term reference pictures are ordered starting with the picture with the highest PicNum value and proceeding through in descending order to the picture with the lowest PicNum value, excluding any picture with a temporal_id value higher than that of the current picture.

The long-term reference pictures are ordered starting with the picture with the lowest LongTermPicNum value and proceeding through in ascending order to the picture with the highest LongTermPicNum value, excluding any picture with a temporal_id value higher than that of the current picture.

For example, when three reference pictures are marked as "used for short-term reference" with PicNum equal to 300, 302, and 303 and two reference pictures are marked as "used for long-term reference" with LongTermPicNum equal to 0 and 3, the initial index order is:

- RefPicList0[0] is set equal to the short-term reference picture with PicNum = 303,
- RefPicList0[1] is set equal to the short-term reference picture with PicNum = 302,
- RefPicList0[2] is set equal to the short-term reference picture with PicNum = 300,
- RefPicList0[3] is set equal to the long-term reference picture with LongTermPicNum = 0,
- RefPicList0[4] is set equal to the long-term reference picture with LongTermPicNum = 3.

8.2.3.3.8.2.2.2.2 Initialisation process for reference picture lists for B slices

This initialisation process is invoked when decoding a B slice in a coded picture.

For purposes of the formation of the reference picture lists RefPicList0 and RefPicList1 the term reference entry refers in the following to decoded reference pictures.

When this process is invoked, there shall be at least one reference entry that is currently marked as "used for reference" (i.e., as "used for short-term reference" or "used for long-term reference") and is not marked as "non-existing". Pictures with higher values of temporal_id than the current picture cannot be used for reference, and are not included in either RefPicList0 or RefPicList1.

For B slices, the order of short-term reference entries in the reference picture lists RefPicList0 and RefPicList1 depends on output order, as given by PicOrderCnt( ). When pic_order_cnt_type is equal to 0, reference pictures that are marked as "non-existing" as specified in subclause 8.2.3.2 are not included in either RefPicList0 or RefPicList1.

NOTE 1 – When gaps_in_frame_num_value_allowed_flag is equal to 1, encoders should use reference picture list modification to ensure proper operation of the decoding process (particularly when pic_order_cnt_type is equal to 0, in which case PicOrderCnt( ) is not inferred for "non-existing" pictures).

The reference picture list RefPicList0 is ordered such that short-term reference entries have lower indices than long-term reference entries. It is ordered as follows:

1. Let entryShortTerm be a variable ranging over all reference entries that are currently marked as "used for short-term reference" and which have a value of temporal_id equal to or lower than the temporal_id of the current picture. When some values of entryShortTerm are present having PicOrderCnt(entryShortTerm) less than PicOrderCnt(CurrPic), these values of entryShortTerm are placed at the beginning of refPicList0 in descending order of PicOrderCnt(entryShortTerm). All of the remaining values of entryShortTerm (when present) are then appended to refPicList0 in ascending order of PicOrderCnt(entryShortTerm).
2. The long-term reference entries which have a value of temporal_id equal to or lower than the temporal_id of the current picture are ordered starting with the long-term reference entry that has the lowest LongTermPicNum value and proceeding through in ascending order to the long-term reference entry that has the highest LongTermPicNum value.

The reference picture list RefPicList1 is ordered so that short-term reference entries have lower indices than long-term reference entries. It is ordered as follows:

1. Let entryShortTerm be a variable ranging over all reference entries that are currently marked as "used for short-term reference" and which have a value of temporal_id equal to or lower than the temporal_id of the current picture. When some values of entryShortTerm are present having PicOrderCnt(entryShortTerm) greater than PicOrderCnt(CurrPic), these values of entryShortTerm are placed at the beginning of refPicList1 in ascending order of PicOrderCnt(entryShortTerm). All of the remaining values of entryShortTerm (when present) are then appended to refPicList1 in descending order of PicOrderCnt(entryShortTerm).
2. Long-term reference entries which have a value of temporal_id equal to or lower than the temporal_id of the current picture are ordered starting with the long-term reference picture that has the lowest LongTermPicNum

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000353

value and proceeding through in ascending order to the long-term reference entry that has the highest LongTermPicNum value.

3. When the reference picture list RefPicList1 has more than one entry and RefPicList1 is identical to the reference picture list RefPicList0, the first two entries RefPicList1[0] and RefPicList1[1] are switched.

8.2.2.3 Modification process for reference picture lists

After the invocation of this process, there shall be no reference pictures with greater temporal_id than the current slice included in the output RefPicList0 or RefPicList1.

When ref_pic_list_modification_flag_l0 is equal to 1, the following applies:

1. Let refIdxL0 be an index into the reference picture list RefPicList0. It is initially set equal to 0.
2. The corresponding syntax elements modification_of_pic_nums_idc are processed in the order they occur in the bitstream. For each of these syntax elements, the following applies.
   - If modification_of_pic_nums_idc is equal to 0 or equal to 1, the process specified in subclause 8.2.2.3.1 is invoked with refIdxL0 as input, and the output is assigned to refIdxL0.
   - Otherwise, if modification_of_pic_nums_idc is equal to 2, the process specified in subclause 8.2.2.3.2 is invoked with refIdxL0 as input, and the output is assigned to refIdxL0.
   - Otherwise (modification_of_pic_nums_idc is equal to 3), the modification process for reference picture list RefPicList0 is finished.

When the current slice is a B slice and ref_pic_list_modification_flag_l1 is equal to 1, the following applies:

1. Let refIdxL1 be an index into the reference picture list RefPicList1. It is initially set equal to 0.
2. The corresponding syntax elements modification_of_pic_nums_idc are processed in the order they occur in the bitstream. For each of these syntax elements, the following applies.
   - If modification_of_pic_nums_idc is equal to 0 or equal to 1, the process specified in subclause ~~8.2.2.3.1~~8.2.2.3.1 is invoked with refIdxL1 as input, and the output is assigned to refIdxL1.
   - Otherwise, if modification_of_pic_nums_idc is equal to 2, the process specified in subclause 8.2.2.3.2 is invoked with refIdxL1 as input, and the output is assigned to refIdxL1.
   - Otherwise (modification_of_pic_nums_idc is equal to 3), the modification process for reference picture list RefPicList1 is finished.

8.2.2.3.1 Modification process of reference picture lists for short-term reference pictures

Input to this process is an index refIdxLX (with X being 0 or 1).

Output of this process is an incremented index refIdxLX.

The variable picNumLXNoWrap is derived as follows.

- If modification_of_pic_nums_idc is equal to 0,

```
if( picNumLXPred − ( abs_diff_pic_num_minus1 + 1 ) < 0 )
    picNumLXNoWrap = picNumLXPred − ( abs_diff_pic_num_minus1 + 1 ) + MaxPicNum      (8-4)
else
    picNumLXNoWrap = picNumLXPred − ( abs_diff_pic_num_minus1 + 1 )
```

- Otherwise (modification_of_pic_nums_idc is equal to 1),

```
if( picNumLXPred + ( abs_diff_pic_num_minus1 + 1 ) >= MaxPicNum )
    picNumLXNoWrap = picNumLXPred + ( abs_diff_pic_num_minus1 + 1 ) − MaxPicNum      (8-5)
else
    picNumLXNoWrap = picNumLXPred + ( abs_diff_pic_num_minus1 + 1 )
```

picNumLXPred is the prediction value for the variable picNumLXNoWrap. When the process specified in this subclause is invoked the first time for a slice (that is, for the first occurrence of modification_of_pic_nums_idc equal to 0 or 1 in the ref_pic_list_modification( ) syntax), picNumL0Pred and picNumL1Pred are initially set equal to CurrPicNum. After each assignment of picNumLXNoWrap, the value of picNumLXNoWrap is assigned to picNumLXPred.

The variable picNumLX is derived as specified by the following pseudo-code:

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000354

```
if( picNumLXNoWrap > CurrPicNum )
    picNumLX = picNumLXNoWrap − MaxPicNum                    (8-6)
else
    picNumLX = picNumLXNoWrap
```

picNumLX shall be equal to the PicNum of a reference picture that is marked as "used for short-term reference" and shall not be equal to the PicNum of a short-term reference picture that is marked as "non-existing". The short-term reference picture with PicNum equal to picNumLX shall not have greater temporal_id than the current slice.

The following procedure is conducted to place the picture with short-term picture number picNumLX into the index position refIdxLX, shift the position of any other remaining pictures to later in the list, and increment the value of refIdxLX.

```
for( cIdx = num_ref_idx_lX_active_minus1 + 1; cIdx > refIdxLX; cIdx□− )
    RefPicListX[ cIdx ] = RefPicListX[ cIdx − 1]
RefPicListX[ refIdxLX++ ] = short-term reference picture with PicNum equal to picNumLX
nIdx = refIdxLX
for( cIdx = refIdxLX; cIdx <= num_ref_idx_lX_active_minus1 + 1; cIdx++ )    (8-7)
    if( PicNumF( RefPicListX[ cIdx ] ) != picNumLX )
        RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ]
```

where the function PicNumF(RefPicListX[cIdx]) is derived as follows.

– If the picture RefPicListX[cIdx] is marked as "used for short-term reference", PicNumF(RefPicListX[cIdx]) is the PicNum of the picture RefPicListX[cIdx].

– Otherwise (the picture RefPicListX[cIdx] is not marked as "used for short-term reference"), PicNumF(RefPicListX[cIdx]) is equal to MaxPicNum.

NOTE 1 – A value of MaxPicNum can never be equal to picNumLX.

NOTE 2 – Within this pseudo-code procedure, the length of the list RefPicListX is temporarily made one element longer than the length needed for the final list. After the execution of this procedure, only elements 0 through num_ref_idx_lX_active_minus1 of the list need to be retained.

8.2.2.3.2 Modification process of reference picture lists for long-term reference pictures

Input to this process is an index refIdxLX (with X being 0 or 1).

Output of this process is an incremented index refIdxLX.

The following procedure is conducted to place the picture with long-term picture number long_term_pic_num into the index position refIdxLX, shift the position of any other remaining pictures to later in the list, and increment the value of refIdxLX.

```
for( cIdx = num_ref_idx_lX_active_minus1 + 1; cIdx > refIdxLX; cIdx□− )
    RefPicListX[ cIdx ] = RefPicListX[ cIdx − 1]
RefPicListX[ refIdxLX++ ] = long-term reference picture with LongTermPicNum equal to long_term_pic_num
nIdx = refIdxLX
for( cIdx = refIdxLX; cIdx <= num_ref_idx_lX_active_minus1 + 1; cIdx++ )    (8-8)
    if( LongTermPicNumF( RefPicListX[ cIdx ] ) != long_term_pic_num )
        RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ]
```

where the long-term reference picture with LongTermPicNum equal to long_term_pic_num shall not have greater temporal_id than the current slice, and the function LongTermPicNumF(RefPicListX[cIdx]) is derived as follows.

– If the picture RefPicListX[cIdx] is marked as "used for long-term reference", LongTermPicNumF(RefPicListX[cIdx]) is the LongTermPicNum of the picture RefPicListX[cIdx].

– Otherwise (the picture RefPicListX[cIdx] is not marked as "used for long-term reference"), LongTermPicNumF(RefPicListX[cIdx]) is equal to 2 * (MaxLongTermFrameIdx + 1).

NOTE 1 – A value of 2 * (MaxLongTermFrameIdx + 1) can never be equal to long_term_pic_num.

NOTE 2 – Within this pseudo-code procedure, the length of the list RefPicListX is temporarily made one element longer than the length needed for the final list. After the execution of this procedure, only elements 0 through num_ref_idx_lX_active_minus1 of the list need to be retained.

8.2.2.4 Mapping process for reference picture lists combination in B slices

[Ed.: (WJ) needs to be checked once again. Try to find better way to represent]

This initialisation process is invoked when decoding a B slice header.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000355

Input to this process are the reference picture list RefPicListX and num_ref_idx_lX_active_minus1 with X being 0 or 1.

Outputs of this process are arrays PredLCToPredLx and RefIdxLCToRefIdxLx.

When the current slice is a B slice and ref_pic_list_modification_flag_lc is equal to 0, the following ordered steps apply:

1. Let refIdxL0 and refIdxL1 be indices into the reference picture lists RefPicListL0 and RefPicListL1. They are initially set equal to 0.
2. Let refIdxLC be an index into PredLCToPredLx and RefIdxLCToRefIdxLx. It is initially set equal to 0.
3. The following process is repeated until refIdxL0 and refIdxL1 are both greater than num_ref_idx_l0_active_minus1 and num_ref_idx_l1_active_minus1, respectively:
   - If refIdxL0 is less than or equal to num_ref_idx_l0_active_minus1,
     - If the entry RefPicListL0[refIdxL0] is the first occurance of the reference picture,

       PredLCToPredLx[refIdxLC] = Pred_L0, (8-9)
       RefIdxLCToRefIdxLx[refIdxLC++] = refIdxL0.

     - refIdxL0++.
   - If refIdxL1 is less than or equal to num_ref_idx_l1_active_minus1 and ref_pic_list_combination_flag equal to 1,
     - If the entry RefPicListL1[refIdxL1] is the first occurance of the reference picture,

       PredLCToPredLx[refIdxLC] = Pred_L1, (8-10)
       RefIdxLCToRefIdxLx[refIdxLC++] = refIdxL1.

     - refIdxL1++.

When the current slice is a B slice and ref_pic_list_modification_flag_lc is equal to 1, the following ordered steps apply:

1. Let refIdxLC be an index into the reference picture list PredLCToPredLx and RefIdxLCToRefIdxLx. It is initially set equal to 0.
2. The corresponding syntax elements pic_from_list_0_flag and ref_idx_list_curr are processed in the order they occur in the bitstream. For each of these syntax elements pairs, the following applies.
   - If pic_from_list_0_flag is equal to 1,

     PredLCToPredLx[refIdxLC] = Pred_L0, (8-11)

   - Otherwise,

     PredLCToPredLx[refIdxLC] = Pred_L1 (8-12)

   - RefIdxLCToRefIdxLx[refIdxLC++] = ref_idx_list_curr

When refIdxLC is greater than num_com_ref_list_active_minus1+ 1, the extra entries past position num_com_ref_list_active_minus1 are discarded from PredLCToPredLx and RefIdxLCToRefIdxLx.

When refIdxLC is less than num_com_ref_list_active_minus1 + 1, the remaining entries in PredLCToPredLx and RefIdxLCToRefIdxLx are set equal to Pred_L0 and 0, respectively.

8.2.3 Decoded reference picture marking process

This process is invoked for decoded pictures when nal_ref_idc is not equal to 0.

NOTE 1 – The decoding process for gaps in frame_num that is specified in subclause 8.2.3.2 may also be invoked when nal_ref_idc is equal to 0, as specified in clause 0.

A decoded picture with nal_ref_idc not equal to 0, referred to as a reference picture, is marked as "used for short-term reference" or "used for long-term reference". A picture that is marked as "used for short-term reference" is identified by its FrameNum. A picture that is marked as "used for long-term reference" is identified by its LongTermFrameIdx.

Pictures marked as "used for short-term reference" or as "used for long-term reference" can be used as a reference for inter prediction when decoding a picture until the picture is marked as "unused for reference".

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000356

A picture can be marked as "unused for reference" by the sliding window reference picture marking process, a first-in, first-out mechanism specified in subclause 8.2.3.3 or by the adaptive memory control reference picture marking process, a customised adaptive marking operation specified in subclause 8.2.3.4.

A short-term reference picture is identified for use in the decoding process by its variables FrameNum and FrameNumWrap and its picture number PicNum, and a long-term reference picture is identified for use in the decoding process by its long-term picture number LongTermPicNum. When the current picture is not an IDR picture, subclause 8.2.2.1 is invoked to specify the assignment of the variables FrameNum, FrameNumWrap, PicNum and LongTermPicNum.

8.2.3.1 Sequence of operations for decoded reference picture marking process

Decoded reference picture marking proceeds in the following ordered steps:

4. All slices of the current picture are decoded.
5. Depending on whether the current picture is an IDR picture, the following applies.
   - If the current picture is an IDR picture, the following ordered steps are specified:
     a. All reference pictures are marked as "unused for reference"
     b. Depending on long_term_reference_flag, the following applies.
        - If long_term_reference_flag is equal to 0, the IDR picture is marked as "used for short-term reference" and MaxLongTermFrameIdx is set equal to "no long-term picture indices".
        - Otherwise (long_term_reference_flag is equal to 1), the IDR picture is marked as "used for long-term reference", the LongTermFrameIdx for the IDR picture is set equal to 0, and MaxLongTermFrameIdx is set equal to 0.
   - Otherwise (the current picture is not an IDR picture), the following applies.
     - If adaptive_ref_pic_marking_mode_flag is equal to 0, the process specified in subclause 8.2.3.3 is invoked.
     - Otherwise (adaptive_ref_pic_marking_mode_flag is equal to 1), the process specified in subclause 8.2.3.4 is invoked.
6. When the current picture is not an IDR picture and it was not marked as "used for long-term reference" by memory_management_control_operation equal to 6, it is marked as "used for short-term reference".

It is a requirement of bitstream conformance that, after marking the current decoded reference picture, the total number of pictures marked as "used for reference" shall not be greater than Max(max_num_ref_frames, 1).

8.2.3.2 Decoding process for gaps in frame_num

[Ed. (TW): insert text]

8.2.3.3 Sliding window decoded reference picture marking process

This process is invoked when adaptive_ref_pic_marking_mode_flag is equal to 0.

Depending on the properties of the current picture as specified below, the following applies.

1. Let numShortTerm be the total number of reference pictures that are marked as "used for short-term reference". Let numLongTerm be the total number of reference pictures that are marked as "used for long-term reference".
2. When numShortTerm + numLongTerm is equal to Max(max_num_ref_frames, 1), the condition that numShortTerm is greater than 0 shall be fulfilled, and the short-term reference picture that has the smallest value of FrameNumWrap is marked as "unused for reference".

8.2.3.4 Adaptive memory control decoded reference picture marking process

This process is invoked when adaptive_ref_pic_marking_mode_flag is equal to 1.

The memory_management_control_operation commands with values of 1 to 6 are processed in the order they occur in the bitstream after the current picture has been decoded. For each of these memory_management_control_operation commands, one of the processes specified in subclauses 8.2.3.4.1 to 8.2.3.4.6 is invoked depending on the value of

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000357

memory_management_control_operation. The memory_management_control_operation command with value of 0 specifies the end of memory_management_control_operation commands.

8.2.3.4.1 Marking process of a short-term reference picture as "unused for reference"

This process is invoked when memory_management_control_operation is equal to 1.

Let picNumX be specified by

$$picNumX = CurrPicNum - (difference\_of\_pic\_nums\_minus1 + 1). \quad (8\text{-}13)$$

The value of picNumX is used to mark the corresponding short-term reference picture as "unused for reference".

8.2.3.4.2 Marking process of a long-term reference picture as "unused for reference"

This process is invoked when memory_management_control_operation is equal to 2.

The value of LongTermPicNum is used to mark the corresponding long-term reference picture as "unused for reference".

8.2.3.4.3 Assignment process of a LongTermFrameIdx to a short-term reference picture

This process is invoked when memory_management_control_operation is equal to 3.

Given the syntax element difference_of_pic_nums_minus1, the variable picNumX is obtained as specified in subclause 8.2.3.4.1. picNumX shall refer to a picture marked as "used for short-term reference" and not marked as "non-existing".

When LongTermFrameIdx equal to long_term_frame_idx is already assigned to a long-term reference picture, that picture is marked as "unused for reference".

The value of LongTermFrameIdx is used to mark the corresponding picture from "used for short-term reference" to "used for long-term reference".

8.2.3.4.4 Decoding process for MaxLongTermFrameIdx

This process is invoked when memory_management_control_operation is equal to 4.

All pictures for which LongTermFrameIdx is greater than max_long_term_frame_idx_plus1 − 1 and that are marked as "used for long-term reference" are marked as "unused for reference".

The variable MaxLongTermFrameIdx is derived as follows.

- If max_long_term_frame_idx_plus1 is equal to 0, MaxLongTermFrameIdx is set equal to "no long-term picture indices".
- Otherwise (max_long_term_frame_idx_plus1 is greater than 0), MaxLongTermFrameIdx is set equal to max_long_term_frame_idx_plus1 − 1.

NOTE – The memory_management_control_operation command equal to 4 can be used to mark long-term reference pictures as "unused for reference". The frequency of transmitting max_long_term_frame_idx_plus1 is not specified by this Recommendation | International Standard. However, the encoder should send a memory_management_control_operation command equal to 4 upon receiving an error message, such as an intra refresh request message.

8.2.3.4.5 Marking process of all reference pictures as "unused for reference" and setting MaxLongTermFrameIdx to "no long-term picture indices"

This process is invoked when memory_management_control_operation is equal to 5.

All reference pictures are marked as "unused for reference" and the variable MaxLongTermFrameIdx is set equal to "no long-term picture indices".

8.2.3.4.6 Process for assigning a long-term picture index to the current picture

This process is invoked when memory_management_control_operation is equal to 6.

When a variable LongTermFrameIdx equal to long_term_frame_idx is already assigned to a long-term reference picture, that picture is marked as "unused for reference".

The current picture is marked as "used for long-term reference" and assigned LongTermFrameIdx equal to long_term_frame_idx.

After marking the current decoded reference picture, the total number of pictures that are marked as "used for reference" shall not be greater than Max(max_num_ref_frames, 1).

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000358

NOTE – Under some circumstances, the above statement may impose a constraint on the order in which a memory_management_control_operation syntax element equal to 6 can appear in the decoded reference picture marking syntax relative to a memory_management_control_operation syntax element equal to 1, 2, 3, or 4.

8.3 Decoding process for coding units coded in intra prediction mode

Inputs to this process are:

– a luma location (xB, yB) specifying the top-left luma sample of the current coding unit relative to the top-left luma sample of the current picture,

– a variable log2CUSize specifying the size of the current coding unit.

Output of this process is:

– a modified reconstructed picture before deblocking filtering.

A variable nS is set equal to (1 << log2CUSize).

Depending on pcm_flag and IntraSplitFlag, the decoding process for luma samples is specified as follows.

– If pcm_flag is equal to 1, the reconstructed picture is modified as follows:

$$recSample_L[\,xB+i,\,xB+j\,] = pcm\_sample\_luma[\,nS*j+i\,] << (\,BitDepth_Y - PCMBitDepth_Y\,) \text{ with } i,j = 0..nS-1 \quad (8\text{-}1443)$$

– Otherwise (pcm_flag is equal to 0), ~~If~~ if IntraSplitFlag is equal to 0, the following ordered steps apply:

1. The derivation process for the intra prediction mode as specified in subclause 오류! 참조 원본을 찾을 수 없습니다. ~~8.3.1~~ is invoked with the luma location (xB, yB) and the variable log2TrafoSize set equal to log2CUSize as well as IntraPredMode that is previously (in decoding order) derived for adjacent coding units as the input and the output is the variable IntraPredMode[xB][yB].

~~2.~~ The decoding process for intra blocks as specified in subclause 8.3.3 is invoked with the luma location (xB, yB), the variable log2TrafoSize set equal to log2CUSize, the variable trafoDepth set equal to 0, the luma intra prediction mode IntraPredMode[xB][yB] and the variable cIdx set equal to 0 as the inputs and the output is a modified reconstructed picture before deblocking filtering.

– Otherwise (pcm_flag is equal to 0 and IntraSplitFlag is equal to 1), for the variable blkIdx proceeding over the values 0..3, the following ordered steps apply:

1. The variable xBS is set equal to xB + (~~(1 << (log2CUSize – 1))~~ nS >> 1) * (blkIdx % 2).
2. The variable yBS is set equal to yB + (~~(1 << (log2CUSize – 1))~~ nS >> 1) * (blkIdx / 2).
3. The derivation process for the intra prediction mode as specified in subclause 오류! 참조 원본을 찾을 수 없습니다. ~~8.3.1~~ is invoked with the luma location (xBS, yBS) and the variable log2TrafoSize set equal to log2CUSize – 1 as well as IntraPredMode that is previously (in decoding order) derived for adjacent coding units as the input and the output is the variable IntraPredMode[xBS][yBS].
4. The decoding process for intra blocks as specified in subclause 8.3.3 is invoked with the luma location (xBS, yBS), the variable log2TrafoSize set equal to log2CUSize – 1, the variable trafoDepth set equal to 1, the luma intra prediction mode IntraPredMode[xBS][yBS] and the variable cIdx set equal to 0 as the inputs and the output is a modified reconstructed picture before deblocking filtering.

Depending on pcm_flag, ~~The~~ the decoding process for chroma samples is specified as follows ~~in the following ordered steps~~:

– If pcm_flag is equal to 1, the reconstructed picture is modified as follows:

$$recSamples_C[\,xB/2+i,\,yB/2+j\,] = pcm\_sample\_chroma[\,nS/2*j+i\,] << (\,BitDepth_C - PCMBitDepth_C\,) \text{ with } i,j = 0..nS/2-1 \quad (8\text{-}1543)$$

$$recSamples_C[\,xB/2+i,\,yB/2+j\,] = pcm\_sample\_chroma[\,(nS/2*(j+nS))+i\,] << (\,BitDepth_C - PCMBitDepth_C\,) \text{ with } i,j = 0..nS/2-1 \quad (8\text{-}1643)$$

– Otherwise (pcm_flag is equal to 0), for the variable blkIdx proceeding over the values 0..3, the following ordered steps apply:

서식 있음: 아래 첨자
서식 있음: 탭: 5.67 글자, 왼쪽
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 들여쓰기: 왼쪽: 0.5 cm, 내어쓰기: 4.25 글자, 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm
서식 있음: 탭: 5.67 글자, 왼쪽

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000359

1. The derivation process for the chroma intra prediction mode as specified in 8.3.2 is invoked with the luma location (xB, yB) as input and the output is the variable IntraPredModeC.
2. The decoding process for intra blocks as specified in subclause 8.3.3 is invoked with the chroma location (xB/2, yB/2), the variable log2TrafoSize set equal to log2CUSize-1, the variable trafoDepth set equal to 0, the chroma intra prediction mode IntraPredModeC, and the variable cIdx set equal to 1 as the inputs and the output is a modified reconstructed picture before deblocking filtering.
3. The decoding process for intra blocks as specified in subclause 8.3.3 is invoked with the chroma location (xB/2, yB/2), the variable log2TrafoSize set equal to log2CUSize-1, the variable trafoDepth set equal to 0, the chroma intra prediction mode IntraPredModeC, and the variable cIdx set equal to 2 as the inputs and the output is a modified reconstructed picture before deblocking filtering.

8.3.1 Derivation process for luma intra prediction mode

Inputs to this process are:

- a luma location (xB, yB) specifying the top-left luma sample of the current block relative to the top-left luma sample of the current picture,
- a variable log2TrafoSize specifying the size of the current prediction unit,
- variable arrays IntraPredMode (If available) that are previously (in decoding order) derived for adjacent coding units.

Output of this process is the variable IntraPredMode[xB][yB].

Table 8-1~~Table 8-5~~ specifies the value for the intra prediction mode and the associated names.

Table **8-15** – Specification of intra prediction mode and associated names

Intra prediction mode	Associated names
0	Intra_Vertical
1	Intra_Horizontal
2	Intra_DC
Otherwise (3..34)	Intra_Angular
34	Intra_Planar
35	Intra_FromLuma (used only for chroma)

Table 8-2~~Table 8-1~~ specifies the number of luma intra prediction modes intraPredModeNum depending on log2TrafoSize.

Table 8-2~~15~~8-1~~3~~ – Specification of intraPredModeNum

log2TrafoSize	intraPredModeNum
2	18~~7~~
3	3~~435~~
4	3~~4~~1~~8~~
5	3~~435~~
6	3~~4~~

Table 8-3~~Table 8-2~~ specifies the mapping table used for converting the number of intra prediction modes.

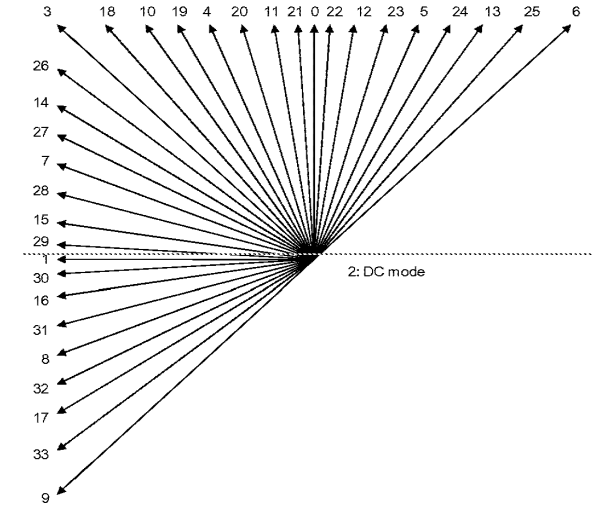
Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000360

Table 8-328-28-3 – Specification of ~~mapIntraPredMode8~~ mapIntraPredMode3 and mapIntraPredMode9

value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
mapIntraPredMode35[value]	0	1	2	~~3~~2	~~4~~2	0	2	1	1	2	2	0	0	2	2	1	1
mapIntraPredMode9[value]	0	1	2	3	4	5	6	7	8	2	2	2	2	2	2	2	2
value	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
mapIntraPredMode35[value]	2	~~3~~2	~~4~~2	~~4~~2	0	0	0	2	2	2	2	1	1	1	1	2	2
mapIntraPredMode9[value]	2	3	4	4	0	0	5	5	6	3	7	7	1	1	8	8	2

IntraPredMode[xB][yB] labelled 0, 1, 2, .., 33 represents directions of predictions as illustrated in ~~Figure 8-1~~Figure 8-1.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000361



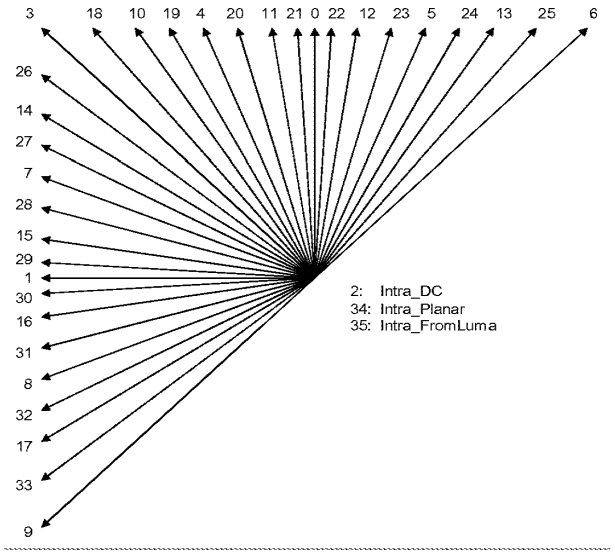


Figure 8-1 – Intra prediction mode directions (informative)

IntraPredMode[xB][yB] is derived as ~~follows~~the following ordered steps. [Ed. (WJ): proponent suggests to move this part to the syntax since the other syntax elements utilize IntraPredMode. But it seems too complex to move all the following process to the syntax table. Maybe it's better to move this part to the semantics section or simply avoid the use of IntraPredMode to parse the syntax item]

1. The derivation process for neighbouring treeblocks specified in subclause XXX with (xB, yB) given as input and the output is assigned to tbAddrA and tbAddrB specifying the treeblock addresses of treeblocks

서식 있음: 들여쓰기: 왼쪽: 0 cm, 첫 줄: 0 cm, 탭: 0 글자, 왼쪽 + 1.42 글자(없음)

서식 있음: 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0.63 cm + 들여쓰기 위치: 1.27 cm, 탭: 3.55 글자, 왼쪽 + 3.97 글자(없음)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000362

covering (xBA, yBA) and (xBB, yBB) respectively where (xBA, yBA) is set equal to (xB-1, yB) and (xBB, yBB) is set equal to (xB, yB-1).

2. For N being either replaced A or B, the variables intraPredModeN are derived as follows.

- If the treeblock with address tbAddrN is not available, intraPredModeN is set equal to -1
- Otherwise, if the coding unit covering (xBN, yBN) is not coded as intra mode, intraPredModeN is set equal to Intra_DC
- Otherwise, intraPredModeN is set equal to IntraPredMode[xBN][yBN], where IntraPredMode is the variable array assigned to the coding unit covering the luma location (xBN, yBN).

3. For N being either replaced A or B, the variables intraPredModeN are further modified to be equal to Intra_DC when intraPredModeN is equal to Intra_Planar.

4. For N being either replaced A or B, the variables candIntraPredModeN are derived as follows.

- If intraPredModeN is greater than or equal to intraPredModeNum
  - If intraPredModeNum is equal to 5 then candIntraPredModeN is set equal to mapIntraPredMode5[intraPredModeN]
  - if intraPredModeNum is equal to 3 then candIntraPredModeN is set equal to mapIntraPredMode3[intraPredModeN]
  - Otherwise candIntraPredModeN is set equal to mapIntraPredMode9[intraPredModeN].
- Otherwise, candIntraPredModeN is set equal to intraPredModeN

5. The candModeList[x] and NumMPMCand are derived as follows:

- If both candIntraPredModeN are not available, then the value 2 is assigned to candModeList[0] and NumMPMCand is set equal to 1
- Otherwise, if only one candIntraPredModeN is available or if both candIntraPredModeN are the same, then this candIntraPredModeN is assigned to candModeList[0] and NumMPMCand is set equal to 1
- Otherwise, both candIntraPredModeN are assigned to the candidate modes list with the smaller of the two candidates at candModeList[0] and the larger at candModeList[1] and NumMPMCand is set equal to 2.

6. IntraPredMode[xB][yB] is derived by applying the following procedure.

- If prev_intra_pred_flag[xB][yB] is true, the IntraPredMode[xB][yB] is set equal to candModeList[mpm_idx[xB][yB]]
- Otherwise IntraPredMode[xB][yB] is derived by applying the following procedure
  - IntraPredMode[xB][yB] = rem_intra_pred_mode
  - for (cIdx = 0; cIdx < NumMPMCand; cIdx++)
    - if (IntraPredMode[xB][yB]>= candModeList[cIdx]) IntraPredMode[xB][yB]++

7. When IntraPredModeY[xB][yB] is equal to Intra_DC and planar_flag_luma is equal to 1, IntraPredMode[xB][yB] is further modified to be equal to Intra_Planar.

8.3.2 Derivation process for chroma intra prediction mode

[Ed.: (WJ) this subclause may be moved to the semantics of intra_chroma_pred_mode syntax]

Input to this process is a luma location (xB, yB) specifying the top-left luma sample of the current block relative to the top-left luma sample of the current picture.

Output of this process is the variable IntraPredModeC.

The chroma intra prediction mode IntraPredModeC is derived as specifed in Table 8-4Table 8-3Table 8-3 or Table 8-5Table 8-4 with intra_chroma_pred_mode, and IntraPredMode[xB][yB] and chroma_pred_from_luma_enabled_flag as inputs.

IntraPredModeC is further modified to be equal to Intra_Planar when IntraPredModeC is equal to Intra_DC and planar_flag_chroma is equal to 1.

서식 있음: 들여쓰기: 왼쪽: 1.29 cm, 탭: 5.67 글자, 왼쪽 + 3.55 글자(없음) + 5.4 글자

서식 있음: 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0.63 cm + 들여쓰기 위치: 1.27 cm, 탭: 1.42 글자, 왼쪽 + 3.55 글자, 왼쪽 + 5.96 글자, 왼쪽 + 7.94 글자, 왼쪽 + 9.93 글자, 왼쪽 + 5.4 글자(없음) + 7.2 글자 + 14.89 글자

서식 있음: (한글) 한국어

서식 있음: 들여쓰기: 왼쪽: 1.29 cm, 탭: 5.67 글자, 왼쪽 + 3.55 글자(없음) + 5.4 글자

서식 있음: 들여쓰기: 왼쪽: 2 cm

서식 있음: 들여쓰기: 왼쪽: 1.29 cm, 탭: 5.67 글자, 왼쪽 + 3.55 글자(없음) + 5.4 글자

서식 있음: 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0.63 cm + 들여쓰기 위치: 1.27 cm, 탭: 1.42 글자, 왼쪽 + 3.55 글자, 왼쪽 + 5.96 글자, 왼쪽 + 7.94 글자, 왼쪽 + 9.93 글자, 왼쪽

서식 있음: 들여쓰기: 왼쪽: 1.29 cm, 탭: 5.67 글자, 왼쪽 + 3.55 글자(없음) + 5.4 글자

서식 있음: 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0.63 cm + 들여쓰기 위치: 1.27 cm, 탭: 3.55 글자, 왼쪽 + 3.97 글자(없음)

서식 있음: (한글) 한국어

서식 있음: 들여쓰기: 왼쪽: 1.29 cm, 탭: 5.67 글자, 왼쪽 + 14.89 글자, 왼쪽 + 3.6 글자(없음) + 5.4 글자

서식 있음: (한글) 한국어

서식 있음: 들여쓰기: 왼쪽: 2 cm

서식 있음: 들여쓰기: 왼쪽: 1.27 cm, 탭: 1.42 글자, 왼쪽 + 3.55 글자, 왼쪽 + 3.97 글자(없음)

서식 있음: 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0.63 cm + 들여쓰기 위치: 1.27 cm, 탭: 1.42 글자, 왼쪽 + 3.55 글자, 왼쪽 + 3.97 글자(없음)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000363

Table 8-4~~38-3~~ – Specification of IntraPredModeC according to the values of intra_chroma_pred_mode and IntraPredMode[xB][yB] when chroma_pred_from_luma_enabled_flag is equal to 1

intra_chroma_pred_mode	IntraPredMode[xB][yB]				
	0	1	2	3	X (4 <= X < ~~34~~35)
0	~~0~~34	~~1~~34	~~2~~34	~~3~~34	X~~34~~
1	~~1~~n/a	0	0	0	0
2	~~2~~1	~~2~~n/a	1	1	1
3	~~3~~2	~~3~~2	~~2~~n/a	2	2
4	~~n/a~~0	~~n/a~~1	~~n/a~~2	~~n/a~~3	~~3~~X

서식 있음: 강조

서식 있음: 캡션

Table ~~8-54~~ – Specification of IntraPredModeC according to the values of intra_chroma_pred_mode and IntraPredMode[xB][yB] when chroma_pred_from_luma_enabled_flag is equal to 0

intra_chroma_pred_mode	IntraPredMode[xB][yB]				
	0	1	2	3	X (4 <= X < 34)
0	n/a	0	0	0	0
1	1	n/a	1	1	1
2	2	2	~~n/a~~2	2	2
3	0	1	2	3	X

서식 있음: 강조

[Ed. (WJ): is it possible that intra_chroma_pred_mode = 2 and IntraPredMode[xB][yB] = 2? Since it is actualy overlapped with the case that intra_chroma_pred_mode = 3 (derivation from luma) and IntraPredMode[xB][yB] = 2]

서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 0 글자, 왼쪽 + 4.26 글자(없음) + 5.67 글자 + 7.09 글자

8.3.3 Decoding process for intra blocks

Inputs to this process are:

- a ~~luma~~ sample location (xB, yB) specifying the top-left sample of the current block relative to the top-left sample of the current picture,
- a variable log2TrafoSize specifying the size of the current block,
- a variable trafoDepth specifying the hierarchy depth of the current block relative to the coding unit,
- a variable intraPredMode specifying the intra prediction mode.
- a variable cIdx specifying the chroma component of the current block,

Output of this process is:

- a modified reconstructed picture before deblocking filtering.

Depending split_transform_flag[xB][yB][trafoDepth], the following applies:

- If split_transform_flag[xB][yB][trafoDepth] is equal to 1, the following ordered steps apply:
  1. The variable xB1 is set equal to xB + ((1 << log2TrafoSize) >> 1).
  2. The variable yB1 is set equal to yB + ((1 << log2TrafoSize) >> 1).
  3. The decoding process for intra blocks as specified in this subclause is invoked with the location (xB, yB), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode intraPredMode, and the variable cIdx as the inputs and the output is a modified reconstructed picture before deblocking filtering.
  4. The decoding process for intra blocks as specified in this subclause is invoked with the location (xB1, yB), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode intraPredMode, and the variable cIdx as the inputs and the output is a modified reconstructed picture before deblocking filtering.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000364

5. The decoding process for intra blocks as specified in this subclause is invoked with the location (xB, yB1), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode intraPredMode, and the variable cIdx as the inputs and the output is a modified reconstructed picture before deblocking filtering.
6. The decoding process for intra blocks as specified in this subclause is invoked with the location (xB1, yB1), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode intraPredMode, and the variable cIdx as the inputs and the output is a modified reconstructed picture before deblocking filtering.

– Otherwise (split_transform_flag[xB][yB][trafoDepth] is equal to 0), the following ordered steps apply:

1. The variable nS is set equal to 1 << log2TrafoSize.
2. The intra sample prediction process as specified in subclause 8.3.3.1 is invoked with the location (xB, yB), the intra prediction mode intraPredMode, the prediction size nS and the variable cIdx as the inputs and the output is a (nS)x(nS) array predSamples.
3. The scaling and transformation process as specified in subclause 8.5.1 is invoked with the location (xB, yB), the variable trafoDepth, the variable cIdx, and the transform size trafoSize set equal to nS as the inputs and the output is a (nS)x(nS) array resSamples.
4. The residual signal accumulation process as specified in subclause XXX is invoked with the variable arraySize set equal to nS, the (nS)x(nS) array predSamples, and the (nS)x(nS) array resSamples as the inputs and the output is a (nS)x(nS) array recSamples.
5. The picture reconstruction process for a component before deblocking filtering as specified in subclause XXX is invoked with the location (xB, yB), the variable arraySize set equal to nS, the variable cIdx set equal to 0, and the (nS)x(nS) array recSamples as the inputs and the output is a modified reconstructed picture before deblocking filtering.

8.3.3.1 Intra sample prediction

Inputs to this process are:

– a sample location (xB, yB) specifying the top-left sample of the current block relative to the top-left sample of the current picture,
– a variable intraPredMode specifying the luma intra prediction mode,
– a variable nS specifying the prediction size.
– a variable cIdx specifying the chroma component of the current block,

Output of this process is:

– the predicted samples predSamples[x, y], with x, y =0..nS-1.

The nS*4+1 neighbouring samples p[x, y] that are constructed luma samples prior to the deblocking filter process, with x = -1, y = -1..nS*2-1 and x = 0..nS*2-1, y=-1, are derived as follows.

– The luma location (xBN, yBN) is specified by

xBN = xB + x (8-17~~14~~)

yBN = yB +y (8-18~~15~~)

– The derivation process for neighbouring locations in subclause XXX is invoked for sample location with (xBN, yBN) as input and the treeblock address tbAddrN as output.
– Each sample p[x, y] with x = -1, y= -1..nS*2-1 and x = 0, y = -1 is derived as follows
  – If any of the following condition is true, the sample p[x, y] is marked as "not available for intra prediction"
    – tbAddrN is not available
    – the coding unit covering (xBN, yBN) is not coded as intra mode and constrained_intra_pred_flag is equal to 1
  – Otherwise, the sample p[x, y] is marked as "available for intra prediction" and the sample at the location (xBN, yBN) inside the treeblock tbAddrN is assigned to p[x, y]

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000365

When intraPredMode is equal to 2, if at least one sample p[x, y] with x = -1, y = 0..nS-1 and x = 0..nS-1, y = -1 is marked as "not available for intra prediction" or cIdx is not equal to 0, DCPredFilterFlag is set equal to 0. Otherwise, DCPredFilterFlag is set equal to 1.

[Ed. (WJ): reference sample subtitution process eliminates the needs to check the availability of samples except DC mode. Maybe it's better to remove this checking even for DC mode for the consistency]

When at least one sample p[x, y] with x = 1, y = 1..nS*2 1 and x = 0..nS*2 1, y = 1 is marked as "not available for intra prediction," the reference sample substitution process for intra sample prediction in subclause 8.3.3.1.1 is invoked with the samples p[x, y] with x = 1, y = 1..nS*2 1 and x = 0..nS*2 1, y = 1 as input and the modified samples p[x, y] with x = 1, y = 1..nS*2 1 and x = 0..nS*2 1, y = 1 as output.

Table x-4 specifies the value for intraPredMode and the associated names.

Table x-88 – Specification of IntraPredMode and associated names

IntraPredMode	Name of IntraPredMode
0	Intra_Vertical
1	Intra_Horizontal
2	Intra_DC
Otherwise (3..33)	Intra_Angular
34	Intra_FromLuma (used only for chroma)

서식 있음: 영어(미국)

Depending on intraPredMode, the following ordered steps apply:

1. When cIdx is equal to 0, filtering process of neighbouring samples specified in 8.3.3.1.2 is invoked with the sample array p and the prediction size nS as the inputs and the output is reassigned to the sample array p[x][y] with x, y = -1..2*nS-1 filtered sample array pF.

2. Intra sample prediction process according to intraPredMode applies as follows:

   - Otherwise, Intra_FromLuma and Otherwise, if intraPredMode is equal to 34, Intra_FromLuma prediction process specified in subclause 8.3.3.1.8 is invoked with the sample location (xB, yB) = (0, 0), the sample array p and the prediction size nS as the inputs and the output is the predicted sample array predSamples.

   3. One of the intra prediction modes specified in subclause 8.3.3.1.3 (8.3.3.1.2) to 8.3.3.1.8 (8.3.3.1.6) is invoked with the sample array p[][] and the prediction size nS as the inputs and the output is the predicted sample array predSamples according to intraPredMode.

서식 있음: 들여쓰기: 왼쪽: 1.25 cm, 내어쓰기: 4.25 글자, 글머리 기호 + 수준:2 + 맞춤 위치: 1.9 cm + 탭 간격: 2.54 cm + 들여쓰기 위치: 2.54 cm, 탭: 5.67 글자, 왼쪽 + 5.4 글자(없음)

8.3.3.1.1 Reference sample substitution process for intra sample prediction

서식 있음: 제목 5, 글머리 기호 또는 번호 없이, 탭: 5.4 글자(없음) + 7.2 글자 + 14.89 글자

Inputs to this process are the reference samples p[x, y] with x = -1, y = -1..nS*2-1 and x = 0..nS*2-1, y = -1 for intra sample prediction.

Outputs of this process are the modified reference samples p[x, y] with x = -1, y = -1..nS*2-1 and x = 0..nS*2-1, y = -1 for intra sample prediction.

The values of the samples p[x, y] with x = -1, y = -1..nS*2-1 and x = 0..nS*2-1, y = -1 are modified as follows:

- If all samples p[x, y] with x = -1, y = -1..nS*2-1 and x = 0..nS*2-1, y = -1 are marked as "not available for intra prediction," the value (1 << ($BitDepth_Y$ - 1)) is substituted for the values of all samples p[x, y].
- Otherwise (at least one but not all samples p[x, y] are marked as "not available for intra prediction"), the following steps are performed for each sample p[x0, y0] with x0 = -1, y0 = -1..nS*2-1 and x0 = 0..nS*2-1, y0 = -1 marked as "not available for intra prediction":
  - Let the variables q and r be initially set to -1.
  - The variable q is modified as follows:
    - If x0 is equal to -1, a reference sample p[x, y] marked as "available for intra prediction" is identified by searching sequentially among p[x, y] starting from x = -1, y = y0+1 to x = -1, y = nS*2-1. As soon as a sample marked as "available for intra prediction" is found, the search is terminated and the value of p[x, y] is assigned to q. [Ed.: (WJ) is there any elegant way rather than searching?]

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000366

- Otherwise (x0 is not equal to -1), a sample marked as "available for intra prediction" is identified among the samples p[x, y] by searching sequentially starting from x = x0-1, y = -1 to x = -1, y = -1. When a sample marked as "available for intra prediction" is not found among p[x, y] with x = x0-1..-1, y = -1, the search is continued among p[x, y] sequentially starting from x = -1, y = 0 to x = -1, y = nS*2-1. As soon as a sample marked as "available for intra prediction" is found, the search is terminated and the value of p[x, y] is assigned to q. [Ed.: (WJ) is there any elegant way rather than searching?]

- The variable r is modified as follows:
  - If x0 is equal to -1, a sample marked as "available for intra prediction" is identified among the samples p[x, y] by searching sequentially starting from x = -1, y = y0-1 to x = -1, y = -1. When a sample marked as "available for intra prediction" is not found among p[x, y] with x = -1, y = y0-1..-1, the search is continued among p[x, y] sequentially starting from x = 0, y = -1 to x = nS*2-1, y = -1. As soon as a sample marked as "available for intra prediction" is found, the search is terminated and the value of p[x, y] is assigned to r. [Ed.: (WJ) is there any elegant way rather than searching?]
  - Otherwise (x0 is not equal to -1), a sample marked as "available for intra prediction" is identified among the samples p[x, y] by searching sequentially starting from x = x0+1, y = -1 to x = nS*2-1, y = -1. As soon as a sample marked as "available for intra prediction" is found, the search is terminated and the value of p[x, y] is assigned to r. [Ed.: (WJ) is there any elegant way rather than searching?]
- The value of the sample p[x0, y0] is modified as follows:
  - If q is not equal to -1 and r is not equal to -1, the value ((q + r + 1) >> 1) is substituted for the value of p[x0, y0].
  - Otherwise, if q is not equal to -1, the value of q is substituted for the value of p[x0, y0].
  - Otherwise (q is equal to -1 and r is not equal to -1), the value of r is substituted for the value of p[x0, y0].

All samples p[x, y] with x = -1, y = -1..nS*2-1 and x = 0..nS*2-1, y = -1 are marked as "available for intra prediction."

Filtering process of neighbouring samples

Inputs to this process are:

- neighbouring samples p[x, y], with x, y = -1..2*nS-1,
- a variable nS specifying the prediction size.

Output of this process is:

- filtered samples pF[x, y],. with x, y = -1..2*nS-1.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000367

Table 8-6~~8-8~~ – Specification of intraFilterType[nS][IntraPredMode] for various prediction unit sizes

IntraPredMode	intraFilterType for nS = 4	intraFilterType for nS = 8	intraFilterType for nS = 16	intraFilterType for nS = 32	intraFilterType for nS = 64
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	1	1	~~2~~1	~~2~~1	~~2~~1
4	0	1	~~2~~1	~~2~~1	~~2~~1
5	0	~~2~~1	~~2~~1	~~2~~1	0
6	1	~~2~~1	~~2~~1	~~2~~1	0
7	0	~~2~~1	~~2~~1	~~2~~1	0
8	0	1	~~2~~1	~~2~~1	0
9	1	1	~~2~~1	~~2~~1	0
10	0	0	~~2~~1	~~2~~1	0
11	0	0	~~2~~1	~~2~~1	0
12	0	0	~~2~~1	~~2~~1	0
13	0	0	~~2~~1	~~2~~1	0
14	0	0	~~2~~1	~~2~~1	0
15	0	0	~~2~~1	~~2~~1	0
16	0	0	~~2~~1	~~2~~1	0
17	0	0	~~2~~1	~~2~~1	0
18-33	0	0	0	~~2~~1	0
~~34~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~
~~35~~	~~n/a~~	~~n/a~~	~~n/a~~	~~n/a~~	~~n/a~~

Filtered sample array pF[x, y] with x = -1..nS*2-1 and y = -1..nS*2-1 are derived as follows:

– ~~If~~ When intraFilterType[nS][IntraPredMode] is equal to 1, the following applies:

pF[-1, nS*2-1] = p[-1, nS*2-1] (8-~~19~~17)

pF[nS*2-1, -1] = p[nS*2-1, -1] (8-~~20~~18)

pF[-1, y] = (p[-1, y+1] + 2*p[-1, y] + p[-1, y-1] + 2) >> 2 for y = nS*2-2..0 (8-~~21~~19)

pF[-1, -1] = (p[-1, 0] + 2*p[-1, -1] + p[0, -1] + 2) >> 2 (8-~~22~~20)

pF[x, -1] = (p[x-1, -1] + 2*p[x, -1] + p[x+1, -1] + 2) >> 2 for x = 0..nS*2-2 (8-~~23~~24)

~~Otherwise, intraFilterType[nS][IntraPredMode] is equal to 2, the following applies:~~

~~pF[-1, nS*2-1] = p[-1, nS*2-1] (8-22)~~

~~pF[nS*2-1, -1] = p[nS*2-1, -1] (8-23)~~

~~pF[-1, y] = (p[-1, y+1] + 2*p[-1, y] + p[-1, y-1] + 2) >> 2 for y = nS*2-2..0 (8-24)~~

~~pF[-1, -1] = (p[-1, 0] + 2*p[-1, -1] + p[0, -1] + 2) >> 2 (8-25)~~

~~pF[x, -1] = (p[x-1, -1] + 2*p[x, -1] + p[x+1, -1] + 2) >> 2 for x = 0..nS*2-2 (8-26)~~

~~pF[-1, y] = (pF[-1, y+1] + 2*pF[-1, y] + pF[-1, y-1] + 2) >> 2 for y = nS*2-2..0 (8-27)~~

서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 내어쓰기: 2.84 글자, 탭: 1.42 글자, 왼쪽 + 4.26 글자(없음) + 5.67 글자 + 7.09 글자

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000368

~~pF[-1, -1] = (pF1[-1, 0] + 2*pF1[-1, -1] + pF1[0, -1] + 2) >> 2 (8-28)~~

~~pF[x, -1] = (pF1[x-1, -1] + 2* pF1[x, -1] + pF1[x+1, -1] + 2) >> 2 for x = 0..nS*2-2 (8-29)~~

~~8.3.3.3.38.3.3.3.3~~ **Specification of Intra_Vertical prediction mode**

Inputs to this process are:

– neighbouring samples p[x, y], with x, y = -1..2*nS-1,

– a variable nS specifying the prediction size.

Output of this process is:

– predicted samples predSamples[x, y], with x, y =0..nS-1.

This intra prediction mode is invoked when intraPredMode is equal to 0.

The values of the prediction samples predSamples[x, y], with x, y = 0..nS-1, are derived by

predSamples[x, y] = p[x, -1], with x, y = 0..nS-1 (8-2430)

~~8.3.3.3.38.3.3.3.4~~ **Specification of Intra_Horizontal prediction mode**

Inputs to this process are:

– neighbouring samples p[x, y], with x, y = -1..2*nS-1,

– a variable nS specifying the prediction size.

Output of this process is:

– predicted samples predSamples[x, y], with x, y =0..nS-1.

This intra prediction mode is invoked when intraPredMode is equal to 1.

The values of the prediction samples predSamples[x, y], with x, y = 0..nS-1, are derived by

predSamples[x, y] = p[-1, y], with x, y = 0..nS-1 (8-2531)

~~8.3.3.3.38.3.3.3.5~~ **Specification of Intra_DC prediction mode**

Inputs to this process are:

– neighbouring samples p[x, y], with x, y = -1..2*nS-1,

– a variable nS specifying the prediction block size.,

~~– the bit depth of the chroma component, bitDepth.~~

Output of this process is:

– predicted samples predSamples[x, y], with x, y =0..nS-1.

This intra prediction mode is invoked when intraPredMode is equal to 2.

The values of the prediction samples predSamples[x, y], with x, y = 0..nS-1, are derived as ~~follows:the following ordered steps:~~

~~If all samples p[x, -1], with x = 0..nS-1 and p[-1, y], with y = 0..nS-1 are marked as "available for intra prediction", the value of the prediction sample predSamples[x, y], with x, y = 0..nS-1 are derived the following ordered steps:~~

1. A variable DCVal is derived as:

~~by~~

~~predSamples[x, y]~~DCVal = $\left(\sum_{x'=0}^{nS-1} p[x', -1] + \sum_{y'=0}^{nS-1} p[-1, y'] + nS \right) >> (k+1)$, with x, y = 0..nS-1 (8-2642)

where $k=\log_2(nS)$

서식 있음: 글꼴: 10 pt

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000369

2. Depending on the prediction block size nS, the following applies.

– If DCPredFilterFlag is equal to 1 and nS is less than 32, the following applies.

– If nS is equal to 4, the prediction samples predSamples[x, y] are derived as

predSamples[0, 0] = (3*p[-1, 0] + 2*DCVal + 3*p[0, -1] + 4) >> 3 (8-2732)
predSamples[x, 0] = (3*p[x, -1] + 5*DCVal + 4) >> 3, with x = 1..nS-1 (8-2832)
predSamples[0, y] = (3*p[-1, y] + 5*DCVal + 4) >> 3, with y = 1..nS-1 (8-2932)
predSamples[x, y] = DCVal, with x, y = 1..nS-1 (8-3032)

– Otherwise, if nS is equal to 8, the prediction samples predSamples[x, y] are derived as

predSamples[0, 0] = (1*p[-1, 0] + 2*DCVal + 1*p[0, -1] + 2) >> 2 (8-3132)
predSamples[x, 0] = (1*p[x, -1] + 3*DCVal + 2) >> 2, with x = 1..nS-1 (8-32)
predSamples[0, y] = (1*p[-1, y] + 3*DCVal + 2) >> 2, with y = 1..nS-1 (8-3332)
predSamples[x, y] = DCVal, with x, y = 1..nS-1 (8-3432)

– Otherwise, if nS is equal to 16, the prediction samples predSamples[x, y] are derived as

predSamples[0, 0] = (1*p[-1, 0] + 6*DCVal + 1*p[0, -1] + 4) >> 3 (8-3532)
predSamples[x, 0] = (1*p[x, -1] + 7*DCVal + 4) >> 3, with x = 1..nS-1 (8-3632)
predSamples[0, y] = (1*p[-1, y] + 7*DCVal + 4) >> 3, with y = 1..nS-1 (8-3732)
predSamples[x, y] = DCVal, with x, y = 1..nS-1 (8-3832)

– Otherwise, the prediction samples predSamples[x, y] are derived as

predSamples[x, y] = DCVal, with x, y = 0..nS-1 (8-3932)

~~Otherwise, if all samples p[x, -1], with x = 0..nS-1 are marked as "not available for intra prediction" and all samples p[-1, y], with y = 0..nS-1 are marked as "available for intra prediction," the values of the prediction samples predSamples[x, y], with x, y = 0..nS-1 are derived by~~

~~$$predSamples[x,y] = \left(\sum_{y'=0}^{nS-1} p[-1,y'] + (nS >> 1)\right) >> k, \text{ with } x,y = 0..nS-1 \quad (8\text{-}33)$$~~

~~where k = log2(nS).~~

~~Otherwise, if all samples p[x, -1], with x = 0..nS-1 are marked as "available for intra prediction" and all samples p[-1, y], with y = 0..nS-1 are marked as "not available for intra prediction," the values of the prediction samples predSamples[x, y], with x, y = 0..nS-1 are derived by~~

~~$$predSamples[x,y] = \left(\sum_{x'=0}^{nS-1} p[x',-1] + (nS >> 1)\right) >> k, \text{ with } x,y = 0..nS-1 \quad (8\text{-}34)$$~~

~~where k = log2(nS).~~

~~Otherwise, the values of the prediction samples predSamples[x, y], with x, y = 0..nS-1 are derived by~~

~~predSamples[x, y] = 1 << (BitDepth_Y - 1), with x, y = 0..nS-1 (8-35)~~

~~[Ed: (WJ) current software implementation seems different. It uses division operation and may have a bug when both above and left samples are not available]~~

8.3.3.1.68 8.3.3.1.6 **Specification of Intra_Angular prediction mode**

[Ed: (WJ) provided text from JCTVC-C042 was modified. It still needs to be improved]

Inputs to this process are:

– neighbouring samples p[x, y], with x, y = -1..2*nS-1,

– a variable nS specifying the prediction size.

서식 있음: 표준, 들여쓰기: 왼쪽: 0.54 cm, 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, ... + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm, 탭: 3.6 글자, 왼쪽 + 5.4 글자, 왼쪽 + 7.2 글자, 왼쪽 + 8.51 글자, 왼쪽 + 9.93 글자, 탭 목록 + 2 글자(없음) + 4.26 글자 + 5.67 글자 + 7.09 글자

서식 있음: 영어(영국)

서식 있음: 들여쓰기: 왼쪽: 2.5 cm, 탭: 4.26 글자(없음) + 5.67 글자

서식 있음: 글꼴: 10 pt, 영어(영국)

서식 있음: 영어(영국)

서식 있음: 표준, 들여쓰기: 왼쪽: 1.87 cm, 글머리 기호 + 수준:1 + 맞춤 위치: 1.25 cm + 들여쓰기 위치: 1.89 cm, 탭: 3.6 글자, 왼쪽 + 7.2 글자, 왼쪽 + 8.51 글자, 왼쪽 + 4.26 글자(없음) + 5.67 글자 + 7.09 글자

서식 있음: 영어(영국)

서식 있음: 들여쓰기: 왼쪽: 2.5 cm, 탭: 4.26 글자(없음) + 5.67 글자

서식 있음 ... [47]

서식 있음: 영어(영국)

서식 있음: 표준, 들여쓰기: 왼쪽: 1.87 cm, 글머리 기호 + 수준:1 + 맞춤 위치: 1.25 cm + 들여쓰기 위치: 1.89 cm, 탭: 3.6 글자, 왼쪽 + 7.2 글자, 왼쪽 + 8.51 글자, 왼쪽 + 4.26 글자(없음) + 5.67 글자 + 7.09 글자

서식 있음 ... [48]

서식 있음: 들여쓰기: 왼쪽: 2.5 cm, 탭: 4.26 글자(없음) + 5.67 글자

서식 있음 ... [49]

서식 있음: 영어(영국)

서식 있음: 표준, 글머리 기호 + 수준:1 + 맞춤 위치: 1.25 cm + 들여쓰기 위치: 1.89 cm, 탭: 3.6 글자, 왼쪽 + 5.4 글자, 왼쪽 + 7.2 글자, 왼쪽 + 8.51 글자, 왼쪽 + 4.26 글자(없음) + 5.67 글자 + 7.09 글자

서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 3.6 글자, 왼쪽 + 5.4 글자, 왼쪽 + 7.2 글자, 왼쪽 + 8.51 글자, 왼쪽 + 4.26 글자(없음) + 5.67 글자 + 7.09 글자

서식 있음: 들여쓰기: 왼쪽: 1.5 cm

서식 있음: 아래 첨자

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000370

Output of this process is:

– predicted samples predSamples[x, y], with x, y =0..nS-1.

This intra prediction mode is invoked when intraPredMode is in the range of 2..34 inclusive.

Table 8-7~~Table 8-6~~ specifies the mapping table between intraPredMode and the rearranged intra prediction order intraPredOrder.

Table 8-7~~8-6~~ – Specification of intraPredOrder

intraPredMode	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
intraPredOrder	-	-	-	1	5	13	17	21	29	33	3	7	11	15	19	23	27
intraPredMode	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
intraPredOrder	31	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32

Figure 8-2~~Figure 8-1~~ illustrates the total 33 intra angles and Table 8-8~~Table 8-7~~ specifies the mapping table between intraPredOrder and the angle parameter intraPredAngle.

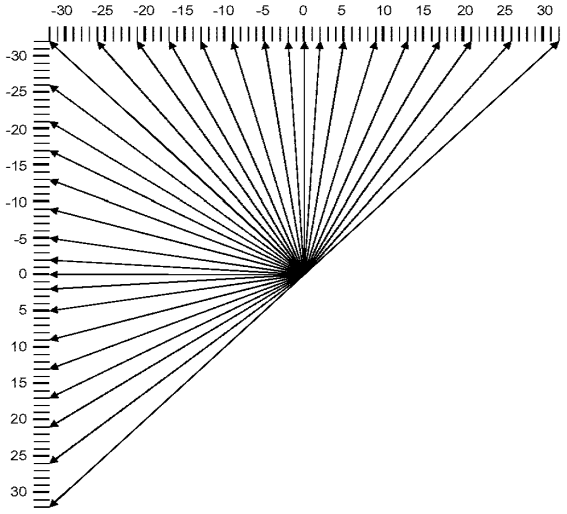


Figure 8-2 – Intra prediction angle definition (informative)

Table 8-8~~8-7~~ – Specification of intraPredAngle

intraPredOrder	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
intraPredAngle	-	-32	-26	-21	-17	-13	-9	-5	-2	-	2	5	9	13	17	21	26
intraPredOrder	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
intraPredAngle	32	-26	-21	-17	-13	-9	-5	-2	-	2	5	9	13	17	21	26	32

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000371

Table 8-9Table 8-8 further specifies the mapping table between intraPredOrder and the inverse angle parameter invAngle.

Table 8-98-88-8 – Specification of invAngle

intraPredOrder	1	2	3	4	5	6	7	8
invAngle	-256	-315	-390	-482	-630	-910	-1638	-4096
intraPredOrder	18	19	20	21	22	23	24	25
invAngle	-315	-390	-482	-630	-910	-1638	-4096	-

The reference pixel array refMain[x], with x=-nS..2*nS is specified as follows.

– If intraPredOrder is less than 18,

refMain[x] = p[-1+x, -1], with x=0..nS (8-4036)

– If intraPredAngle is less than 0,

refMain[x] = p[-1, -1+((x*invAngle+128)>>8)], with x=(nS*intraPredAngle) >>5..-1 (8-4137)

– Otherwise,

refMain[x] = p[-1+x, -1], with x=nS+1..2*nS (8-4238)

Otherwise,

refMain[x] = p[-1, -1+x], with x=0..nS (8-4339)

– If intraPredAngle is less than 0,

refMain[x] = p[-1+((x*invAngle+128)>>8), -1], with x=(nS*intraPredAngle) >>5..-1 (8-4440)

– Otherwise,

refMain[x] = p[-1, -1+x], with x=nS+1..2*nS (8-4541)

The values of the prediction samples predSamples[x, y], with x, y = 0..nS-1 are derived by the following procedures.

– The index variable iIdx and the multiplication factor iFact are derived by

iIdx = ((y + 1)*intraPredAngle) >> 5 (8-4643)

iFact = ((y + 1)*intraPredAngle) && 31 (8-4744)

– Depending on the value of iFact, the following applies.

– If iFact is not equal to 0, the value of the prediction samples predSamples[x, y] is derived by

predSamples[x, y] = ((32 – iFact)*refMain[x+iIdx+1] + iFact*refMain[x+iIdx+2] + 16) >> 5 (8-4844)

– Otherwise, the value of the prediction samples predSamples[x, y] is derived by

predSamples[x, y] = refMain[x+iIdx+1] (8-4945)

8.3.3.1.7 Specification of Intra_Planar prediction mode

Inputs to this process are:

– neighbouring samples p[x, y], with x, y = -1, 2*nS-1,

– a variable nS specifying the prediction size,

– the bit depth of the chroma component, bitDepth.

서식 있음: 제목 5, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000372

Output of this process is:

– predicted samples predSamples[x, y], with x, y = 0..nS-1.

This intra prediction mode is invoked when intraPredMode is equal to 34.

The modified p[x, y] and condVar[x, y] arrays with x, y = 0..nS-1 are initialized as follows.

- If all samples p[x, -1], with x = 0..nS-1 and p[-1, y], with y = 0..nS-1 are marked as "available for intra prediction," the initialization is done as specified for p[x, y] and condVar[x, y] as p[x, y] and condVar[x, y] as p[x, -1].
- Otherwise, if all samples p[x, -1], with x = 0..nS-1 are marked as "not available for intra prediction" and all samples p[-1, y], with y = 0..nS-1 are marked as "available for intra prediction," the initialization is done as specified for p[x, -1] and condVar[x, -1].
- Otherwise, if all samples p[x, -1], with x = 0..nS-1 are marked as "available for intra prediction" and all samples p[-1, y], with y = 0..nS-1 are marked as "not available for intra prediction," the initialization is done as specified for p[-1, y] and condVar[-1, y].
- Otherwise, the initialization is done as specified for p[x, y] = 1 << (BitDepth_Y – 1) and condVar[x, y] = (1 << (BitDepth_Y – 1))².

The values of the prediction samples predSamples[x, y], with x, y = 0..nS-1, are derived by

$$\begin{aligned} predSamples[x,y] = {} & (nS - 1 - x) * condVar[-1, y] + (x + 1) * condVar[nS - 1, -1] + \\ & (nS - 1 - y) * condVar[x, -1] + (y + 1) * condVar[-1, nS - 1] + nS) >> (k + 1) \end{aligned} \quad (8\text{-}5045)$$

with x, y = 0..nS-1, where k = log2(nS).

8.3.3.1.8 Specification of IntraFromLumaprediction mode

Inputs to this process are:

– a sample location (xB, yB) specifying the top-left sample of the current block relative to the top-left sample of the current picture,

– neighbouring samples p[x, y], with x, y = -1, 2*nS-1,

– a variable nS specifying the prediction size.

Output of this process is:

– predicted samples predSamples[x, y], with x, y = 0..nS-1.

This intra prediction mode is invoked when intraPredMode is equal to 34.

The values of the prediction samples predSamples[x, y], with x, y = 0..nS-1, are derived as the following ordered steps:

1. Variable k3 and the sample array p_Y' are derived as:

$$k3 = Max(0, BitDepth_C - \log_2(nS) - 14) \quad (8\text{-}5130)$$

$$p_Y'[x, y] = (recSamples_L[2x, 2y] + recSamples_L[2x, 2y+1]) >> 1, \text{ with } x, y = -1, 2*nS \quad (8\text{-}5230)$$

2. Variables L, C, LL, LC and k2 are derived as follows:

- If all samples p[x, -1], with x = 0..nS-1 and p[-1, y], with y = 0..nS-1 are marked as "available for intra prediction," the variables L, C, LL and LC are derived as:

$$L = \left(\sum_{y=0}^{nS-1} p_Y'[-1, y] + \sum_{x=0}^{nS-1} p_Y'[x, -1] \right) >> k3 \quad (8\text{-}5330)$$

$$C = \left(\sum_{y=0}^{nS-1} p[-1, y] + \sum_{x=0}^{nS-1} p[x, -1] \right) >> k3 \quad (8\text{-}5430)$$

$$LL = \left(\sum_{y=0}^{nS-1} p_Y'[-1, y]^2 + \sum_{x=0}^{nS-1} p_Y'[x, -1]^2 \right) >> k3 \quad (8\text{-}5530)$$

서식 있음 ... [50]

서식 있음: 탭: 1.42 글자(없음)

서식 있음 ... [51]

서식 있음 ... [52]

서식 있음 ... [53]

서식 있음: 아래 첨자

서식 있음: 제목 5, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자

서식 있음: 들여쓰기: 왼쪽: 0.54 cm, 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, ... + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm, 탭: 3.6 글자, 왼쪽 + 5.4 글자, 왼쪽 + 7.2 글자, 왼쪽 + 8.51 글자, 왼쪽 + 3.97 글자(없음) + 5.96 글자 + 7.94 글자 + 9.93 글자

서식 있음: 아래 첨자

서식 있음 ... [54]

서식 있음 ... [55]

서식 있음: 들여쓰기: 왼쪽: 0.54 cm, 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, ... + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm, 탭: 3.6 글자, 왼쪽 + 5.4 글자, 왼쪽 + 7.2 글자, 왼쪽 + 8.51 글자, 왼쪽 + 3.97 글자(없음) + 5.96 글자 + 7.94 글자 + 9.93 글자

서식 있음: 들여쓰기: 왼쪽: 1.29 cm, 글머리 기호 + 수준:1 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm, 탭: 5.67 글자, 탭 목록 + 7.2 글자, 왼쪽 + 14.89 글자, 왼쪽 + 2 글자(없음) + 3.97 글자 + 5.96 글자 + 7.94 글자 + 9.93 글자

서식 있음: 글꼴: 10 pt

서식 있음: 글꼴: 10 pt

서식 있음: 글꼴: 10 pt

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000373

$$LC = \left(\sum_{y=0}^{nS-1} p_Y'[-1, y] * p[-1, y] + \sum_{y=0}^{nS-1} p_Y'[x, -1] * p[x, -1] \right) >> k3 \quad (8\text{-}5629)$$

$k2 = \log_2((2*nS) >> k3)$ (8-5739)

(otherwise, if all samples $p[x, y]$, with $x = 0..nS-1$, are marked as "not available for intra prediction" and all samples $p[-1, y]$, with $y = 0..nS-1$, are marked as "available for intra prediction", the variables LC, LL, and LC are derived as:

$$L = \left(\sum_{y=0}^{nS-1} p_Y'[-1, y] \right) >> k3 \quad (8\text{-}30)$$

$$C = \left(\sum_{y=0}^{nS-1} p_C[-1, y] \right) >> k3 \quad (8\text{-}30)$$

$$LL = \left(\sum_{y=0}^{nS-1} p_Y'[-1, y]^2 \right) >> k3 \quad (8\text{-}30)$$

$$LC = \left(\sum_{y=0}^{nS-1} p_Y'[-1, y] * p_C[-1, y] \right) >> k3 \quad (8\text{-}30)$$

$k2 = \log_2(nS >> k3)$ (8-30)

Otherwise, if all samples $p[x, -1]$, with $x = 0..nS-1$, are marked as "available for intra prediction" and all samples $p[-1, y]$, with $y = 0..nS-1$, are marked as "not available for intra prediction", the variables L, C, LL, and LC are derived as:

$$L = \left(\sum_{x=0}^{nS-1} p_Y'[x, -1] \right) >> k3 \quad (8\text{-}30)$$

$$C = \left(\sum_{x=0}^{nS-1} p_C[x, -1] \right) >> k3 \quad (8\text{-}30)$$

$$LL = \left(\sum_{x=0}^{nS-1} p_Y'[x, -1]^2 \right) >> k3 \quad (8\text{-}30)$$

$$LC = \left(\sum_{y=0}^{nS-1} p_Y'[x, -1] * p_C[x, -1] \right) >> k3 \quad (8\text{-}30)$$

$k2 = \log_2(nS >> k3)$. (8-30)

3. Variables a and b are derived as:

$a1 = (LC << k2) - L*C$ (8-5830)
$a2 = (LL << k2) - L*L$ (8-5930)
$k = Max(0, \log_2(abs(a2)) - 5) - Max(0, \log_2(abs(a1)) - 14) + 2$ (8-6030)
$a1s = a1 >> Max(0, \log_2(abs(a1)) - 14)$ (8-6130)
$a2s = abs(a2 >> Max(0, \log_2(abs(a2)) - 5)) + 1$ (8-6230)

[Ed. (WJ): bug in software – a2s should be abs(a2 >> Max(0, $\log_2$(abs(a2)) – 5)) – without "+1"]

서식 있음: 글꼴: 10 pt

서식 있음: 글꼴: 10 pt

서식 있음: 글꼴: 10 pt, 아래 첨자

서식 있음: 글꼴: 10 pt

서식 있음: 글꼴: 10 pt

서식 있음: 글꼴: 10 pt

서식 있음: 표준, 들여쓰기: 왼쪽: 1.29 cm, 글머리 기호 + 수준:1 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm, 탭: 5.67 글자, 탭 목록 + 7.2 글자, 왼쪽 + 14.89 글자, 왼쪽 + 2 글자(없음) + 4.26 글자 + 7.09 글자

서식 있음: 표준, 들여쓰기: 왼쪽: 1.29 cm, 글머리 기호 + 수준:1 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm, 탭: 5.67 글자, 탭 목록 + 7.2 글자, 왼쪽 + 14.89 글자, 왼쪽 + 2 글자(없음) + 4.26 글자 + 7.09 글자

서식 있음: 들여쓰기: 왼쪽: 0.54 cm, 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, ... + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm, 탭: 3.6 글자, 왼쪽 + 5.4 글자, 왼쪽 + 7.2 글자, 왼쪽 + 8.51 글자, 왼쪽 + 3.97 글자(없음) + 5.96 글자 + 7.94 글자 + 9.93 글자

서식 있음: Equation, 들여쓰기: 왼쪽: 1.5 cm, 탭: 4.26 글자, 왼쪽 + 5.67 글자, 왼쪽 + 7.09 글자, 왼쪽

서식 있음: 글꼴: 10 pt, 아래 첨자

서식 있음: 글꼴: 10 pt, 아래 첨자

서식 있음: 글꼴: 10 pt, 아래 첨자

서식 있음: 글꼴: 10 pt, 아래 첨자

서식 있음: 글꼴: 10 pt, 영어(미국)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000374

a = Clip3(-2^15, 2^15-1, a1s*lmDiv + (1 << (k1 – 1)) >> k1) (8-6338)
b = (L – ((a*C) >> k1) + (1 << (k2 – 1))) >> k2 (8-6339)

where lmDiv is specified in Table 8-10 with the input a2s.

4. The values of the prediction samples predSamples[x, y] are derived as:

predSamples[x, y] = Clip1_C(((nW)[x, y] * a) >> 13) + b), with x, y = 0..nS-1 (8-6340)

Table 8-16 – Specification of lmDiv

a2s	1	2	3	4	5	6	7	8	9	10	11	12	13
lmDiv	32768	16384	10923	8192	6554	5461	4681	4096	3641	3277	2979	2731	2521
a2s	14	15	16	17	18	19	20	21	22	23	24	25	26
lmDiv	2341	2185	2048	1928	1820	1725	1638	1560	1489	1425	1365	1311	1260
a2s	27	28	29	30	31	32	33	34	35	36	37	38	39
lmDiv	1214	1170	1130	1092	1057	1024	993	964	936	910	886	862	840
a2s	40	41	42	43	44	45	46	47	48	49	50	51	52
lmDiv	819	799	780	762	745	728	712	697	683	669	655	643	630
a2s	53	54	55	56	57	58	59	60	61	62	63	64	
lmDiv	618	607	596	585	575	565	555	546	537	529	520	512	

8.4 Decoding process for coding units coded in inter prediction mode

Inputs to this process are:

– a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,

– a variable log2CUSize specifying the size of the current coding unit.

Output of this process is a modified reconstructed picture before deblocking filtering.

The variable nCS_L is set equal to 1 << log2CUSize and the variable nCS_C is set equal to (1 << log2CUSize) >> 1.

Decoding process for coding units coded in inter prediction mode consists of following ordered steps:

1. The inter prediction process as specified in subclause 8.4.1 is invoked with the luma location (xC, yC), coding unit size log2CUSize, inter prediction mode PredMode, and prediction partition mode PartMode as the inputs and the outputs are 3 arrays $predSamples_L$, $predSamples_{Cb}$, $predSamples_{Cr}$.
2. The decoding process for the residual signal of coding units coded in inter prediction mode specified in subclause 8.4.3 is invoked with the luma location (xC, yC), the size of the current coding unit log2CUSize as inputs and the outputs are 3 arrays $resSamples_L$, $resSamples_{Cb}$, $resSamples_{Cr}$.
3. The residual signal accumulation process as specified in subclause XXXX is invoked with arrays $predSamples_L$, $predSamples_{Cb}$, $predSamples_{Cr}$, and the arrays $resSamples_L$, $resSamples_{Cb}$, $resSamples_{Cr}$ as the inputs and the outputs are 3 arrays $recSamples_L$, $recSamples_{Cb}$, $recSamples_{Cr}$. [Ed: (WJ) it may be split per each color component. Revisit after writing residual signal accumulation process]
4. The picture reconstruction process for a component before deblocking filtering as specified in subclause XXXX is invoked with the luma location (xB, yB), and 3 arrays recSamplesL, recSamplesCb, recSamplesCr as the inputs and the output is a modified reconstructed picture before deblocking filtering. [Ed: (WJ) it may be split per each color component. Revisit after writing picture reconstruction process]

서식 있음: 글꼴: 10 pt, 위 첨자
서식 있음: 글꼴: 10 pt, 위 첨자
서식 있음: 글꼴: 10 pt
서식 있음: 들여쓰기: 왼쪽: 0.54 cm, 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm, 탭: 3.6 글자, 왼쪽 + 5.4 글자, 왼쪽 + 7.2 글자, 왼쪽 + 8.51 글자, 왼쪽 + 3.97 글자(없음) + 5.96 글자 + 7.94 글자 + 9.93 글자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자
서식 있음: 캡션
서식 있음: (한글) 한국어
서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자
서식 있음: (한글) 한국어, (언어2) (없음)
서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자
서식 있음: 들여쓰기: 왼쪽: 0.54 cm, 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000375

8.4.1 Inter prediction process

This process is invoked when decoding coding unit whose PredMode is not equal to MODE_INTRA.

Inputs to this process are:

- a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,
- a variable log2CUSize specifying the size of the current coding unit,
- a variable PredMode specifying prediction mode of current coding unit,
- a variable PartMode specifying prediction partition mode of current coding unit.

Outputs of this process are:

- a $(nCS_L)x(nCS_L)$ array $predSamples_L$ of luma prediction samples, where nCS_L is derived as specified below,
- a $(nCS_C)x(nCS_C)$ array $predSamples_{Cb}$ of chroma prediction samples for the component Cb, where nCS_C is derived as specified below,
- a $(nCS_C)x(nCS_C)$ array $predSamples_{Cr}$ of chroma prediction samples for the component Cr, where nCS_C is derived as specified below.

The variable nCS_L is set equal to 1 << log2CUSize and the variable nCS_C is set equal to (1 << log2CUSize) >> 1. [Ed: (WJ) revisit for supporting other chroma formats]

The variable $nCS1_L$ is set equal to $nCS_L >> 1$.

Depending on PartMode, the following applies: [Ed: (WJ) is PUType better?]

- If PartMode is equal to PART_2Nx2N, the following ordered steps apply:
  1. The decoding process for prediction units in inter prediction mode as specified in subclause 8.4.2 is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to (0, 0), the size of the coding unit nCS_L, the width of the luma prediction samples nPSW set equal to nCS_L, the height of the luma prediction samples nPSH set equal to nCS_L and a partition index PartIdx set equal to 0 as inputs, and the outputs are a $(nCS_L)x(nCS_L)$ array $predSamples_L$ and two $(nCS_C)x(nCS_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.
- Otherwise, if PartMode is equal to PART_2NxN, the following ordered steps apply:
  1. The decoding process for prediction units in inter prediction mode as specified in subclause 8.4.2 is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to (0, 0), the size of the coding unit nCS_L, the width of the luma prediction samples nPSW set equal to nCS_L, the height of the luma prediction samples nPSH set equal to $nCS1_L$ and a partition index PartIdx set equal to 0 as inputs, and the outputs are a $(nCS_L)x(nCS_L)$ array $predSamples_L$ and two $(nCS_C)x(nCS_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.
  2. The decoding process for prediction units in inter prediction mode as specified in subclause 8.4.2 is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to (0, $nCS1_L$), the size of the coding unit nCS_L, the width of the luma prediction samples nPSW set equal to nCS_L, the height of the luma prediction samples nPSH set equal to $nCS1_L$ and a partition index PartIdx set equal to 1 as inputs, and the outputs are a $(nCS_L)x(nCS_L)$ array $predSamples_L$ and two $(nCS_C)x(nCS_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.

Otherwise, if PartMode is equal to PART_Nx2N, the following ordered steps apply:

1. The decoding process for prediction units in inter prediction mode as specified in subclause 8.4.2 is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to (0, 0), the size of the coding unit nCS_L, the width of the luma prediction samples nPSW set equal to $nCS1_L$, the height of the luma prediction samples nPSH set equal to nCS_L and a partition index PartIdx set equal to 0 as inputs, and the outputs are a $(nCS_L)x(nCS_L)$ array $predSamples_L$ and two $(nCS_C)x(nCS_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.
2. The decoding process for prediction units in inter prediction mode as specified in subclause 8.4.2 is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to ($nCS1_L$, 0), the size of the coding unit nCS_L, the width of the luma prediction samples nPSW set equal to $nCS1_L$, the height of the luma prediction samples nPSH set equal to nCS_L and a partition index PartIdx set equal to 1 as inputs, and the outputs are a $(nCS_L)x(nCS_L)$ array $predSamples_L$ and two $(nCS_C)x(nCS_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.

Otherwise, if PartMode is equal to PART_NxN, the following ordered steps apply:

1. The decoding process for prediction units in inter prediction mode as specified in subclause 8.4.2 is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to (0, 0), the size of the coding unit nCS_L, the width of the luma prediction samples nPSW set equal to $nCS1_L$, the height of the luma prediction

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000376

samples nPSH set equal to $nCS1_L$ as and a partition index PartIdx set equal to 0 inputs, and the outputs are a $(nCS_L)x(nCS_L)$ array $predSamples_L$ and two $(nCS_C)x(nCS_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.

2. The decoding process for prediction units in inter prediction mode as specified in subclause 8.4.2 is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to ($nCS1_L$, 0), the size of the coding unit nCS_L, the width of the luma prediction samples nPSW set equal to $nCS1_L$, the height of the luma prediction samples nPSH set equal to $nCS1_L$ and a partition index PartIdx set equal to 1 as inputs, and the outputs are a $(nCS_L)x(nCS_L)$ array $predSamples_L$ and two $(nCS_C)x(nCS_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.

3. The decoding process for prediction units in inter prediction mode as specified in subclause 8.4.2 is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to (0, $nCS1_L$), the size of the coding unit nCS_L, the width of the luma prediction samples nPSW set equal to $nCS1_L$, the height of the luma prediction samples nPSH set equal to $nCS1_L$ and a partition index PartIdx set equal to 2 as inputs, and the outputs are a $(nCS_L)x(nCS_L)$ array $predSamples_L$ and two $(nCS_C)x(nCS_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.

4. The decoding process for inter prediction units as specified in subclause 8.4.2 is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to ($nCS1_L$, $nCS1_L$), the size of the coding unit nCS_L, the width of the luma prediction samples nPSW set equal to $nCS1_L$, the height of the luma prediction samples nPSH set equal to $nCS1_L$ and a partition index PartIdx set equal to 3 as inputs, and the outputs are a $(nCS_L)x(nCS_L)$ array $predSamples_L$ and two $(nCS_C)x(nCS_C)$ arrays $predSamples_{Cb}$ and $predSamples_{Cr}$.

8.4.2 Decoding process for prediction units in inter prediction mode

Inputs to this process are:

- a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,
- a luma location (xB, yB) specifying the top-left luma sample of the current prediction unit relative to the top left luma sample of the current coding unit,
- a variable nCS specifying the size of the current coding unit,
- a variable nPSW specifying the width of the current prediction unit,
- a variable nPSH specifying the width of the current prediction unit,
- a variable PartIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are:

- a $(nCS_L)x(nCS_L)$ array $predSamples_L$ of luma prediction samples, where nCS_L is derived as specified below,
- a $(nCS_C)x(nCS_C)$ array $predSamples_{Cb}$ of chroma prediction samples for the component Cb, where nCS_C is derived as specified below,
- a $(nCS_C)x(nCS_C)$ array $predSamples_{Cr}$ of chroma prediction samples for the component Cr, where nCS_C is derived as specified below.

The variable nCS_L is set equal to nCS and the variable nCS_C is set equal to nCS >> 1. [Ed: (WJ) revisit for supporting other chroma formats]

The decoding process for prediction units in inter prediction mode consists of the following ordered steps:

1. Derivation process for motion vector components and reference indices as specified in subclause 8.4.2.1.

   Inputs to this process are

   - a luma location (xC, yC) of the top-left luma sample of the current coding unit relative to the top-left luma sample of the current picture,
   - a luma location (xB, yB) of the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current coding unit,
   - variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH,
   - a variable PartIdx specifying the index of the current prediction unit within the current coding unit.

   Outputs of this process are

   - luma motion vectors mvL0 and mvL1, and chroma motion vectors mvCL0 and mvCL1,
   - reference indices refIdxL0 and refIdxL1,

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000377

– prediction list utilization flags predFlagL0 and predFlagL1.

2. Decoding process for inter sample prediction as specified in subclause 8.4.2.2.

Inputs to this process are

– a luma location (xC, yC) of the top-left luma sample of the current coding unit relative to the top-left luma sample of the current picture,

– a luma location (xB, yB) of the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current coding unit,

– a variable nCS specifying the size of the current coding unit,

– variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH.

– luma motion vectors mvL0 and mvL1, and chroma motion vectors mvCL0 and mvCL1,

– reference indices refIdxL0 and refIdxL1,

– prediction list utilization flags predFlagL0 and prefFlagL1.

Outputs of this process are

– inter prediction samples (predSamples); which are a (nCS_L)x(nCS_L) array $predSamples_L$ of prediction luma samples and two (nCS_C)x(nCS_C) arrays $predSamples_{Cr}$, and $predSamples_{Cr}$ of prediction chroma samples, one for each of the chroma components Cb and Cr.

For use in derivation processes of variables invoked later in the decoding process, the following assignments are made:

MvL0[xB, yB] = mvL0 (8-~~66~~46)
MvL1[xB, yB] = mvL1 (8-~~67~~47)

RefIdxL0[xB, yB] = refIdxL0 (8-~~68~~48)
RefIdxL1[xB, yB] = refIdxL1 (8-~~69~~49)

PredFlagL0[xB, yB] = predFlagL0 (8-~~70~~50)
PredFlagL1[xB, yB] = predFlagL1 (8-~~71~~54)

8.4.2.1 Derivation process for motion vector components and reference indices

Input to this process are

– a luma location (xC, yC) of the top-left luma sample of the current coding unit relative to the top-left luma sample of the current picture,

– a luma location (xB, yB) of the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current coding unit,

– variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH,

– a variable PartIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are

– luma motion vectors mvL0 and mvL1 and chroma motion vectors mvCL0 and mvCL1,

– reference indices refIdxL0 and refIdxL1,

– prediction list utilization flags predFlagL0 and predFlagL1.

Let (xP, yP) specify the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current picture where xP = xC + xB and yP = yC + yB.

For the derivation of the variables mvL0 and mvL1, refIdxL0 and refIdxL1 as well as PredFlagL0 and PredFlagL1, the following applies.

– If PredMode is equal to MODE_SKIP, the derivation process for luma motion vectors for merge mode as specified in subclause 8.4.2.1.1~~8.4.2.1.3~~ is invoked with the luma location (xP, yP), variables nPSW, nPSH and the partition index PartIdx as inputs and the output being the luma motion vectors mvL0, mvL1, the reference indices refIdxL0, refIdxL1, and the prediction list utilization flags predFlagL0 and predFlagL1.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000378

– Otherwise, if PredMode is equal to MODE_INTER and merge_flag[xP][yP] is equal to 1,, the derivation process for luma motion vectors for merge mode as specified in subclause 8.4.2.1.1~~8.4.2.1.3~~ is invoked with the luma location (xP, yP), variables nPSW and nPSH and the partition index PartIdx as inputs and the outputs being the luma motion vectors mvL0 and mvL1, the reference indices refIdxL0 and refIdxL1, the prediction utilization flags predFlagL0 and predFlagL1.

– Otherwise, for X being replaced by either 0 or 1 in the variables predFlagLX, mvLX, refIdxLX and in Pred_LX and in the syntax elements ref_idx_lX and mvd_lX, the following applies.

1. The variables LcToLx, refIdxLX and predFlagLX are derived as follows.

   – If inter_pred_flag[xP][yP] is equal to Pred_LC and PredLCToPredLx[ref_idx_lc[xP][yP]] is equal to Pred_LX,

   refIdxLX = RefIdxLCToRefIdxLx[ref_idx_lc[xP][yP]] (8-~~72~~)
   predFlagLX = 1 (8-~~73~~)
   mvd_lX[xP][yP][0] = mvd_lc[xP][yP][0] (8-~~74~~)
   mvd_lX[xP][yP][1] = mvd_lc[xP][yP][1] (8-~~75~~)
   mvp_idx_lX[xP][yP][0] = mvp_idx_lc[xP][yP] (8-~~76~~)
   LcToLx = LX (8-~~77~~)

   – Otherwise, if inter_pred_flag[xP][yP] is equal to Pred_LX or Pred_BI,

   refIdxLX = ref_idx_lX[xP][yP] (8-~~78~~)
   predFlagLX = 1 (8-~~79~~)

   – Otherwise, the variables refIdxLX and predFlagLX are specified by

   refIdxLX = -1 (8-~~80~~)
   predFlagLX = 0 (8-~~81~~)

2. The variable mvdLX is derived as follows.

   mvdLX[0] = mvd_lX[xP][yP][0] (8-~~82~~)
   mvdLX[1] = mvd_lX[xP][yP][1] (8-~~83~~)

3. When predFlagLX is equal to 1, the variable mvpLX is derived as follows.

   – The derivation process for luma motion vector prediction in subclause 8.4.2.1.3~~8.4.2.1.5~~ is invoked with the luma location (xP, yP), variables nPSW and nPSH and refIdxLX as the inputs and the output being mvpLX.

4. When predFlagLX is equal to 1, the luma motion vector mvLX is derived as

   mvLX[0] = mvpLX[0] + mvdLX[0] (8-~~84~~)
   mvLX[1] = mvpLX[1] + mvdLX[1] (8-~~85~~)

When ChromaArrayType is not equal to 0 and predFlagLX (with X being either 0 or 1) is equal to 1, the derivation process for chroma motion vectors in subclause 8.4.2.1.7~~8.4.2.1.8~~ is invoked with mvLX and refIdxLX as inputs and the output being mvCLX.

서식 있음: 제목 5, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자

8.4.2.1.1 Derivation process for luma motion vectors for merge mode

This process is only invoked when PredMode is equal to MODE_ INTER and merge_flag [xP][yP] is equal to 1, where (xP, yP) specify the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current picture.

Inputs of this process are

– a luma location (xP, yP) of the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current picture,

– variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH,

– a variable PartIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are

– the luma motion vectors mvL0 and mvL1,

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000379

– the reference indices refIdxL0 and refIdxL1,

– the prediction list utilization flags predFlagL0 and predFlagL1.

The motion vectors mvL0 and mvL1, the reference indices refIdxL0 and refIdxL1, and the prediction utilization flags predFlagL0 and predFlagL1 are derived as specified by the following ordered steps:

1. The derivation process for merging candidates from neighboring prediction unit partitions in subclause 8.4.2.1.4 is invoked with luma location (xP, yP), the width and the height of the prediction unit nPSW and nPSH and the partition index PartIdx as inputs and the output is assigned to the availability flags availableFlagN, the motion vectors mvL0N and mvL1N, the reference indices refIdxL0N and refIdxL1N and the prediction list utilization flags predFlagL0N and predFlagL1N with N being replaced by A, B, C or D.

2. The derivation process of reference indices for temporal merging candidate in subclause 8.4.2.1.3 is invoked with luma location (xP, yP), nPSW, nPSH as the inputs and the output is directly assigned to refIdxLX.

3. The derivation process for temporal luma motion vector prediction in subclause 8.4.2.1.7 is invoked with luma location (xP, yP), refIdxLX as the inputs and with the output being the availability flag availableFlagLXCol and the temporal motion vector mvLXCol. The variables availableFlagCol and predFlagLXCol (with X being 0 or 1, respectively) are derived as specified below.

$$availableFlagCol = availableFlagL0Col \,||\, availableFlagL1Col \quad (8\text{-}86)$$

$$predFlagLXCol = availableFlagLXCol \quad (8\text{-}87)$$

4. The merging candidate list, mergeCandList, is constructed of which elements are given as specified order:

    1. A, if availableFlagA is equal to 1
    2. B, if availableFlagB is equal to 1
    3. Col, if availableFlagCol is equal to 1
    4. C, if availableFlagC is equal to 1
    5. D, if availableFlagD is equal to 1

5. If several merging candidates have the motion vectors and the same reference indices, the merging candidates are removed from the list except the merging candidate which has the smallest order in the mergeCandList.

6. If the number of elements NumMergeCand within the mergeCandList is equal to 1, mergeIdx is set equal to 0, otherwise, mergeIdx is set equal to merge_idx[xP][yP].

7. The following assignments are made with N being the candidate at position mergeIdx in the merging candidate list mergeCandList (N = mergeCandList[mergeIdx]) and X being replaced by 0 or 1:

$$mvLX[\,0\,] = mvLXN[\,0\,] \quad (8\text{-}88)$$

$$mvLX[\,1\,] = mvLXN[\,1\,] \quad (8\text{-}89)$$

$$refIdxLX = refIdxLXN \quad (8\text{-}90)$$

$$predFlagLX = predFlagLXN \quad (8\text{-}91)$$

8. If all availability flags availableFlagN (with N being replaced by A, B, Col, C, or D) are equal to 0, mergeIdx is set equal to 0 and the variables mvLX, refIdxLX and predFlagLX (with X being replaced by 0 or 1) are inferred as follows.

    If slice_type is equal to P, the following applies.

$$mvLX[\,0\,] = 0 \quad (8\text{-}92)$$

$$mvLX[\,1\,] = 0 \quad (8\text{-}93)$$

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000380

refIdxL0 = 0 (8-94)

refIdxL1 = -1 (8-95)

predFlagL0 = 1 (8-96)

prefFlagL1 = 0 (8-97)

Otherwise (slice_type is equal to B), the following applies.

mvLX[0] = 0 (8-98)

mvLX[1] = 0 (8-99)

refIdxL0 = 0 (8-100)

refIdxL1 = 0 (8-101)

predFlagL0 = 1 (8-102)

prefFlagL1 = 1 (8-103)

8.4.2.1.2 **Derivation process for spatial merging candidates**

Inputs to this process are

- a luma location (xP, yP) specifying the top-left luma sample of the current prediction unit relative to the top-left sample of the current picture,
- variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH,
- a variable PartIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are (with N being replaced by A, B, C, or D and with X being replaced by 0 or 1)

- the availability flags availableFlagN of the neighbouring prediction units,
- the reference indices refIdxLXN of the neighbouring prediction units,
- the prediction list utilization flags predFlagLXN of the neighbouring prediction units,
- the motion vectors mvLXN of the neighbouring prediction units.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000381

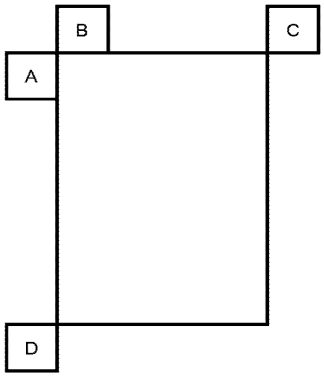


Figure 8-3 – Spatial neighbours that can be used as merging candidates (informative) illustrates the position of the spatial neighbours A, B, C and D relative to the current prediction unit.

For the derivation of availableFlagN, with N being A, B, C or D and (xN, yN) being (xP – 1, yP), (xP, yP – 1), (xP + nPSW, yP – 1) or (xP – 1, yP+nPSH) the following applies.

- If one of the following conditions is true, the availableFlagN is set equal to 0, both components mvLXN are set equal to 0, refIdxLXN and predFlagLX[xN, yN] of the prediction unit covering luma location (xN, yN) are assigned respectively to mvLXN, refIdxLXN and predFlagLXN.
  - The prediction unit covering luma location (xN, yN) is not available or PredMode is MODE_INTRA.
  - PartMode of the current prediction unit is PART_2NxN and PartIdx is equal to 1 and the prediction units covering luma location (xP, yP – 1) (PartIdx = 0) and luma location (xN, yN) (Cand. N) have identical motion parameters:
    - mvLX[xP, yP – 1] = = mvLX[xN, yN]
    - refIdxLX[xP, yP – 1] = = refIdxLX[xN, yN]
    - predFlagLX[xP, yP – 1] = = predFlagLX[xN, yN]
  - PartMode of the current prediction unit is PART_Nx2N and PartIdx is equal to 1 and the prediction units covering luma location (xP – 1, yP) (PartIdx = 0) and luma location (xN, yN) (Cand. N) have identical motion parameters:
    - mvLX[xP – 1, yP] = = mvLX[xN, yN]
    - refIdxLX[xP – 1, yP] = = refIdxLX[xN, yN]
    - predFlagLX[xP – 1, yP] = = predFlagLX[xN, yN]
  - PartMode of the current prediction unit is PART_NxN and PartIdx is equal to 3 and the prediction units covering luma location (xP – 1, yP) (PartIdx = 2) and luma location (xP – 1, yP – 1) (PartIdx = 0) have identical motion parameters:
    - mvLX[xP – 1, yP] = = mvLX[xP – 1, yP – 1]
    - refIdxLX[xP – 1, yP] = = refIdxLX[xP – 1, yP – 1]
    - predFlagLX[xP – 1, yP] = = predFlagLX[xP – 1, yP – 1]

    and the prediction units covering luma location (xP, yP – 1) (PartIdx = 1) and luma location (xN, yN) (Cand. N) have identical motion parameters:
    - mvLX[xP, yP – 1] = = mvLX[xN, yN]
    - refIdxLX[xP, yP – 1] = = refIdxLX[xN, yN]
    - predFlagLX[xP, yP – 1] = = predFlagLX[xN, yN]

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000382

– PartMode of the current prediction unit is PART_NxN and PartIdx is equal to 3 and the prediction units covering luma location (xP, yP – 1) (PartIdx = 1) and luma location (xP – 1, yP – 1) (PartIdx = 0) have identical motion parameters:

  – mvLX[xP, yP – 1] = = mvLX[xP – 1, yP – 1]

  – refIdxLX[xP, yP – 1] = = refIdxLX[xP – 1, yP – 1]

  – predFlagLX[xP, yP – 1] = = predFlagLX[xP – 1, yP – 1]

  and the prediction units covering luma location (xP – 1, yP) (PartIdx = 2) and luma location (xN, yN) (Cand. N) have identical motion parameters:

  – mvLX[xP – 1, yP] = = mvLX[xN, yN]

  – refIdxLX[xP – 1, yP] = = refIdxLX[xN, yN]

  – predFlagLX[xP – 1, yP] = = predFlagLX[xN, yN]

– Otherwise, availableFlagN is set equal to 1 and the variables mvLX[xN, yN], refIdxLX[xN, yN] and predFlagLX[xN, yN] of the prediction unit covering luma location (xN, yN) are assigned respectively to mvLXN, refIdxLXN and predFlagLXN.

8.4.2.1.2 8.4.2.1.3 Derivation process of reference indices for temporal merging candidate

Inputs to this process are

– a luma location (xP, yP) specifying the top-left luma sample of the current prediction unit relative to the top-left sample of the current picture,

– variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH.

Outputs of this process are (with X being replaced by 0 or 1)

– the reference indices refIdxLX of the current prediction units.

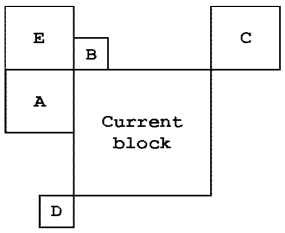


Figure 8-4 – Spatial neighbours, that can be used to derive reference indices for temporal merging candidates, (informative) illustrates the position of the spatial neighbours A, B, C, D and E relative to the current prediction unit.

For the derivation of refIdxLXA, the following applies.

– If the prediction unit covering luma location (xP-1, yP) is available and PredMode is not MODE_INTRA, refIdxLXA is assigned to refIdxLX[xP-1, yP].

– Otherwise, refIdxLXA is assigned to -1.

For the derivation of refIdxLXB, the following applies.

– If the prediction unit covering luma location (xP, yP-1) is available and PredMode is not MODE_INTRA, refIdxLXB is assigned to refIdxLX[xP, yP-1].

– Otherwise, refIdxLXB is assigned to -1.

For the derivation of refIdxLXC, the following applies.

– If the prediction unit covering luma location (xP+nPSW, yP-1) is available and PredMode is not MODE_INTRA, refIdxLXC is assigned to refIdxLX[xP+nPSW, yP-1].

– Otherwise if the prediction unit covering luma location (xP-1, yP+nPSH) is available and PredMode is not MODE_INTRA, refIdxLXC is assigned to refIdxLX[xP-1, yP+nPSH].

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000383

– Otherwise if the prediction unit covering luma location (xP-1, yP-1) is available and PredMode is not MODE_INTRA refIdxLXC is assigned to refIdxLX[xP-1, yP-1].

– Otherwise, refIdxLXC is assigned to -1.

The variable refIdxLX is derived as follows.

– If refIdxLXA is equal to refIdxLXB and refIdxLXB is equal to refIdxLXC, the following applies.
  – If refIdxLXA is equal to -1 then refIdxLX = 0
  – Otherwise refIdxLX = refIdxLXA
– Otherwise if refIdxLXA is equal to refIdxLXB, the following applies.
  – If refIdxLXA is equal to -1 then refIdxLX = refIdxLXC
  – Otherwise refIdxLX = refIdxLXA
– Otherwise if refIdxLXB is equal to refIdxLXC, the following applies.
  – If refIdxLXB is equal to -1 then refIdxLX = refIdxLXA
  – Otherwise refIdxLX = refIdxLXB
– Otherwise if refIdxLXA is equal to refIdxLXC, the following applies.
  – If refIdxLXA is equal to -1 then refIdxLX = refIdxLXB
  – Otherwise refIdxLX = refIdxLXA
– Otherwise if refIdxLXA is equal to -1, the following applies.
  – refIdxLX = min(refIdxLXB, refIdxLXC)
– Otherwise if refIdxLXB is equal to -1, the following applies.
  – refIdxLX = Min(refIdxLXA, refIdxLXC)
– Otherwise if refIdxLXC is equal to -1, the following applies.
  – refIdxLX = Min(refIdxLXA, refIdxLXB)
– Otherwise, the following applies.
  – refIdxLX = Min(Min(refIdxLXA, refIdxLXB), refIdxLXC)

8.4.3.3.38.4.2.1.5 **Derivation process for luma motion vector prediction**

Inputs to this process are

– a luma location (xP, yP) specifying the top-left luma sample of the current prediction unit relative to the top-left sample of the current picture,
– variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH.
– the reference index of the current prediction unit partition refIdxLX (with X being 0 or 1).

Output of this process is

– the prediction mvpLX of the motion vector mvLX (with X being 0 or 1).

The motion vector predictor mvpLX is derived in the following ordered steps.

1. The derivation process for motion vector predictor candidates from neighboring prediction unit partitions in subclause 8.4.2.1.58.4.3.4.6 is invoked with luma location (xP, yP), the width and the height of the prediction unit nPSW and nPSH, and refIdxLX (with X being 0 or 1, respectively) as inputs and the availability flags availableFlagLXN and the motion vectors mvLXN with N being replaced by A, B as the output.
2. The derivation process for temporal luma motion vector prediction in subclause 18.4.2.1.7 is invoked with luma location (xP, yP) , the width and the height of the prediction unit nPSW and nPSH, and refIdxLX (with X being 0 or 1, respectively) as the inputs and with the output being the availability flag availableFlagLXCol and the temporal motion vector predictor mvLXCol.
3. The motion vector predictor list, mvpListLX, is constructed of which elements are given as specified order:
   1. mvLXCol, if availableFlagLXCol is equal to 1

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000384

2. mvLXA, if availableFlagLXA is equal to 1
3. mvLXB, if availableFlagLXB is equal to 1

4. If several motion vectors have the same value, the motion vectors are removed from the list except the motion vector which has the smallest order in the mvpListLX.
5. If the number of elements NumMVPCand(LX) within the mvpListLX is equal to 1, mvpIdx is set equal to 0, otherwise, mvpIdx is set equal to mvp_idx_lX[xP, yP].
6. The motion vector of mvListLX[mvpIdx] is assigned to mvpLX.

8.4.2.1.6.8.2.1.5 Derivation process for motion vector predictor candidates

Inputs to this process are

– a luma location (xP, yP) specifying the top-left luma sample of the current prediction unit relative to the top-left sample of the current picture,

– variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH,

– the reference index of the current prediction unit partition refIdxLX (with X being 0 or 1).

Outputs of this process are (with N being replaced by A,or B)

– the motion vectors mvLXN of the neighbouring prediction units,

– the availability flags availableFlagLXN of the neighbouring prediction units.

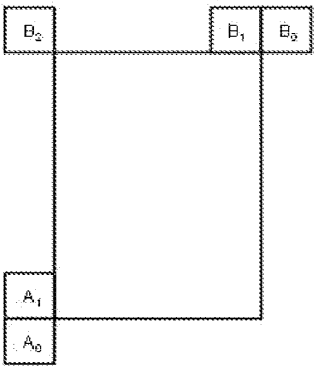


Figure 8-5 – Spatial motion vector neighbours

The function RefPicOrderCnt(pic, refidx, LX) is specified by the value of PicOrderCnt of the picture that is the reference picture RefPicListX[refidx] of pic with X being 0 or 1. PicOrderCnt of the reference picture shall be maintained until the picture is marked as "non-exisiting."

The motion vector mvLXA and the availability flag availableFlagLXA are derived in the following ordered steps:

1. Let a set of two sample locations be (xA_k, yA_k), with k = 0 .. 1, specifies sample locations with $xA_k = xP - 1$, $yA_0 = yP + nPSH$ and $yA_1 = yA_0 - MinPuSize$. The set of sample locations (xA_k, yA_k) represent the sample locations immediately to the left side of the left partition boundary and it's extended line. [Ed.: (WJ) MinPuSize should be defined somewhere]
2. Let the availability flag availableFlagLXA be initially set equal to 0 and the both components of mvLXA are set equal to 0.
3. For (xA_k, yA_k) from (xA_0, yA_0) to (xA_1, yA_1) where $yA_1 = yA_0 - MinPuSize$, the following applies repeatedly until availableFlagLXA is equal to 1:

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000385

– When the prediction unit covering luma location (xA_k, yA_k) is available, PredMode is not MODE_INTRA, predFlagLX[xA_k][yA_k] is equal to 1 and the reference index refIdxLX[xA_k][yA_k] is equal to the reference index of the current prediction unit refIdxLX, availableFlagLXA is set equal to 1 and the motion vector mvLXA is set equal to the motion vector mvLX[xA_k][yA_k], refIdxA is set equal to refIdxLX[xA_k][yA_k] and ListA is set equal to LX.

4. When availableFlagLXA is equal to 0, for (xA_k, yA_k) from (xA_0, yA_0) to (xA_1, yA_1) where $yA_1 = yA_0$ - MinPuSize, the following applies repeatedly until availableFlagLXA is equal to 1:

   – If the prediction unit covering luma location (xAk, yAk) is available, PredMode is not MODE_INTRA, predFlagLY[xA_k][yA_k] (with Y = !X) is equal to 1 and RefPicOrderCnt(currPic, refIdxLY[xA_k][yA_k], LY) is equal to RefPicOrderCnt(currPic, refIdxLX, LX), availableFlagLXA is set equal to 1, the motion vector mvLXA is set equal to the motion vector mvLY[xA_k][yA_k], refIdxA is set equal to refIdxLY[xA_k][yA_k] and ListA is set equal to LY.

   – Otherwise if the prediction unit covering luma location (xA_k, yA_k) is available, PredMode is not MODE_INTRA, predFlagLX[xA_k][yA_k] is equal to 1, availableFlagLXA is set equal to 1, the motion vector mvLXA is set equal to the motion vector mvLX[xA_k][yA_k], refIdxA is set equal to refIdxLX[xA_k][yA_k], ListA is set equal to LX.

   – Otherwise if the prediction unit covering luma location (xA_k, yA_k) is available, PredMode is not MODE_INTRA, predFlagLY[xA_k][yA_k] (with Y = !X) is equal to 1, availableFlagLXA is set equal to 1, the motion vector mvLXA is set equal to the motion vector mvLY[xA_k][yA_k], refIdxA is set equal to refIdxLY[xA_k][yA_k], ListA is set equal to LY.

5. When availableFlagLXA is equal to 1, the following applies.

   – If RefPicOrderCnt(currPic , refIdxA, ListA) is equal to RefPicOrderCnt(currPic, refIdxLX, LX), mvLXA is set equal to mvLXA

   – Otherwise, mvLXA is derived as specified below

$$tx = (16384 + Abs(td / 2)) / td \quad (8\text{-}104??)$$

$$DistScaleFactor = Clip3(-1024, 1023, (tb * tx + 32) >> 6) \quad (8\text{-}105??)$$

$$mvLXA = ClipMv((DistScaleFactor * mvLXA + 128) >> 8) \quad (8\text{-}106??)$$

[Ed. (WJ): software has a clip function for scaled motion vector. Do we need this? AVC does not have it]

where td and tb are derived as

$$td = Clip3(-128, 127, PicOrderCnt(currPic) – RefPicOrderCnt(currPic , refIdxA, ListA)) \quad (8\text{-}107??)$$

$$tb = Clip3(-128, 127, PicOrderCnt(currPic) – RefPicOrderCnt(currPic, refIdxLX, LX)) \quad (8\text{-}108??)$$

The motion vector mvLXB and the availability flag availableFlagLXB are derived in the following ordered steps:

1. Let a set of three sample location (xB_k, yB_k), with k = 0,1,2, specifies sample locations with $xB_0 = xP + nPSW$, $xB_1 = xB_0$ MinPuSize , $xB_2 = xP$ - MinPuSize and $yB_k = yP – 1$. The set of sample locations (xB_k, yB_k) represent the sample locations immediately to the upper side of the above partition boundary and it's extended line. [Ed.: (WJ) MinPuSize should be defined somewhere]

2. Let the availability flag availableFlagLXB be initially set equal to 0 and the both components of mvLXB are set equal to 0.

3. For (xB_k, yB_k) from (xB_0, yB_0) to (xB_2, yB_2) where $xB_0 = xP + nPSW$, $xB_1 = xB_0$ - MinPuSize , and $xB_2 =$ xP - MinPuSize, the following applies repeatedly until availableFlagLXB is equal to 1:

   – When the prediction unit covering luma location (xB_k, yB_k) is available, PredMode is not MODE_INTRA, predFlagLX[xB_k][yB_k] is equal to 1, and the reference index refIdxLX[xB_k][yB_k] is equal to the reference index of the current prediction unit refIdxLX and mvLX[xB_k][yB_k] is not identical to mvLXA, availableFlagLXB is set equal to 1 and the motion vector mvLXB is set equal to the motion vector mvLX[xB_k][yB_k], refIdxB is set equal to refIdxLX[xB_k][yB_k] and ListB is set equal to LX.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000386

4. When availableFlagLXB is equal to 0, for (xB_k, yB_k) from (xB_0, yB_0) to (xB_2, yB_2) where xB_0 = xP +nPSW, xB_1 = xB_0 - MinPuSize , and xB_2 = xP - MinPuSize, the following applies repeatedly until availableFlagLXB is equal to 1:

   – If the prediction unit covering luma location (xB_k, yB_k) is available, PredMode is not MODE_INTRA, predFlagLY[xB_k][yB_k](with Y = !X) is equal to 1 and RefPicOrderCnt(currPic , refIdxLY[xB_k][yB_k], LY) is equal to RefPicOrderCnt(currPic, refIdxLX, LX), availableFlagLXB is set equal to 1, the motion vector mvLXB is set equal to the motion vector mvLY[xB_k][yB_k], refIdxB is set equal to refIdxLY[xB_k][yB_k] and ListB is set equal to LY.

   – Otherwise if the prediction unit covering luma location (xB_k, yB_k) is available, PredMode is not MODE_INTRA, predFlagLX[xB_k][yB_k] is equal to 1, availableFlagLXB is set equal to 1, the motion vector mvLXB is set equal to the motion vector mvLX[xB_k][yB_k], refIdxB is set equal to refIdxLX[xB_k][yB_k], ListB is set equal to LX.

   – Otherwise if the prediction unit covering luma location (xB_k, yB_k) is available, PredMode is not MODE_INTRA, predFlagLY[xB_k][yB_k] (with Y = !X) is equal to 1, availableFlagLXB is set equal to 1, the motion vector mvLXB is set equal to the motion vector mvLY[xB_k][yB_k], refIdxB is set equal to refIdxLY[xB_k][yB_k], ListB is set equal to LY.

   – If availableFlagLXB is equal to 1 and RefPicOrderCnt(currPic , refIdxB, ListB) is equal to RefPicOrderCnt(currPic, refIdxLX, LX), mvLXB is set equal to mvLXB.

   – Otherwise if availableFlagLXB is equal to 1 and RefPicOrderCnt(currPic , refIdxB, ListB) is not equal to RefPicOrderCnt(currPic, refIdxLX, LX),, mvLXB is derived as specified below.

$$tx = (16384 + Abs(td / 2)) / td \quad (8\text{-}10977)$$

$$DistScaleFactor = Clip3(-1024, 1023, (tb * tx + 32) >> 6) \quad (8\text{-}11033)$$

$$mvLXB = ClipMv((DistScaleFactor * mvLXB + 128) >> 8) \quad (8\text{-}11177)$$

[Ed. (WJ): software has a clip function for scaled motion vector. Do we need this? AVC does not have it]

where td and tb are derived as

$$td = Clip3(-128, 127, PicOrderCnt(currPic) – RefPicOrderCnt(currPic , refIdxB, ListB)) \quad (8\text{-}11277)$$

$$tb = Clip3(-128, 127, PicOrderCnt(currPic) – RefPicOrderCnt(currPic, refIdxLX, LX)) \quad (8\text{-}11343)$$

   – When availableFlagLXB is equal to 1 and mvLXB is identical to mvLXA, availableFlagLXB is set equal to 0.

8.4.2.1.8.8.4.2.1.8 **Derivation process for temporal luma motion vector prediction**

Inputs to this process are

– a luma location (xP, yP) specifying the top-left luma sample of the current prediction unit relative to the top-left sample of the current picture,

– variables specifying the width and the height of the prediction unit for luma, nPSW and nPSH,

– the reference index of the current prediction unit partition refIdxLX (with X being 0 or 1).

Outputs of this process are

– the motion vector prediction mvLXCol,

– the availability flag availableFlagLXCol.

The function RefPicOrderCnt(pic, refidx, LX) is specified by the value of PicOrderCnt of the picture that is the reference picture RefPicListX[refidx] of pic with X being 0 or 1. PicOrderCnt of the reference picture shall be maintained until the picture is marked as "non-exisiting."

Depending on the values of slice_type and collocated_from_l0_flag, the variable colPic, specifying the picture that contains the co-located partition, is derived as follows.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000387

– If slice_type is equal to B and collocated_from_l0_flag is equal to 0, the variable colPic specifies the picture that contains the co-located partition as specified by RefPicList1[0].

– Otherwise (slice_type is equal to B and collocated_from_l0_flag is equal to 1 or slice_type is equal to P) , the variable colPic specifies the picture that contains the co-located partition as specified by RefPicList0[0].

Variable colPu and its position (xPCol, yPCol) are derived in the following ordered steps:

1. Right-bottom luma position of the current prediction unit is defined by

$$xPRb = xP + nPSW \quad (8\text{-}114\text{77})$$

$$yPRb = yP + nPSH \quad (8\text{-}115\text{77})$$

2. The variable colPu is set as the prediction unit covering the modified position given by ((xPRb >> 4) << 4, (yPRb >> 4) << 4) inside the colPic.

3. If colPu is coded in an intra prediction mode or colPu is unavailable, the following applies.

   – Central luma position of the current prediction unit is defined by

$$xPCtr = (xP + (nPSW >> 1) – 1 \quad (8\text{-}116\text{77})$$

$$yPCtr = (yP + (nPSH >> 1) – 1 \quad (8\text{-}117\text{78})$$

   – The variable colPu is set as the prediction unit covering the modified position given by ((xPCtr >> 4) << 4, (yPCtr >> 4) << 4) inside the colPic.

4. (xPCol, yPCol) is set equal to the top-left luma sample of the colPu relative to the top-left luma sample of the colPic.

The variables mvLXCol and availableFlagLXCol are derived as follows.

– If colPu is coded in an intra prediction mode or colPu is unavailable, both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.

– Otherwise (colPu is not coded in an intra prediction mode and colPu is available), the variables mvCol and refIdxCol are derived as follows,

– If PredFlagL0[xPCol][yPCol] is equal to 0, the motion vector mvCol and the reference index refIdxCol are set equal to MvL1[xPCol][yPCol] and RefIdxL1[xPCol][yPCol], respectively.

– Otherwise (PredFlagL0[xPCol][yPCol] is equal to 1), the following applies.

  – If PredFlagL1[xPCol][yPCol] is equal to 0, the motion vector mvCol and the reference index refIdxCol are set equal to MvL0[xPCol][yPCol] and RefIdxL0[xPCol][yPCol], respectively.

  – Otherwise (PredFlagL1[xPCol][yPCol] is equal to 1), the following applies.

    a. The following asignments are made with X being 0 or 1.

       – RefIdxColLX = RefIdxLX[xPCol][yPCol]

       – If PicOrderCnt(colPic) is less than PicOrderCnt(currPic) and RefPicOrderCnt(colPic, RefIdxColLX, LX) is greater than PicOrderCnt(currPic) or PicOrderCnt(colPic) is greater than PicOrderCnt(currPic) and RefPicOrderCnt(colPic, RefIdxColLX, LX) is less than PicOrderCnt(currPic), the variable MvXCross is derived as follows.

         – MvXCross = 1

       – Otherwise (PicOrderCnt(colPic) is less than PicOrderCnt(currPic) and RefPicOrderCnt(colPic, RefIdxColLX, LX) is less than or equal to PicOrderCnt(currPic) or PicOrderCnt(colPic) is greater than PicOrderCnt(currPic) and RefPicOrderCnt(colPic, RefIdxColLX, LX) is greater than or equal to PicOrderCnt(currPic)), the variable MvXCross is derived as follows.

         – MvXCross = 0

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000388

b. If one of the following conditions is true, the motion vector mvCol, the reference index refIdxCol and ListCol are set equal to MvL1[xPCol][yPCol], RefIdxColL1 and L1, respectively.

- Mv0Cross is equal to 0 and Mv1Cross is equal to 1
- Mv0Cross is equal to Mv1Cross and reference index list is equal to L1

c. Otherwise, the motion vector mvCol, the reference index refIdxCol and ListCol are set equal to MvL0[xPCol][yPCol], RefIdxColL0 and L0, respectively.

and the variable availableFlagLXCol is set equal to 1 and the following applies.

- If PicOrderCnt(colPic) – RefPicOrderCnt(colPic, refIdxCol, ListCol) is equal to PicOrderCnt(currPic) - RefPicOrderCnt(currPic, refIdxLX, LX),

mvLXCol = mvCol (8-11879)

- Otherwise, mvLXCol is derived as scaled version of the motion vector mvCol as specified below

tx = (16384 + Abs(td / 2)) / td (8-11980)

DistScaleFactor = Clip3(-1024, 1023, (tb * tx + 32) >> 6) (8-12024)

mvLXCol = ClipMv((DistScaleFactor * mvCol + 128) >> 8) (8-12182)

[Ed. (WJ): software has a clip function for scaled motion vector. Do we need this? AVC does not have it]

where td and tb are derived as

td = Clip3(-128, 127, PicOrderCnt(colPic) – RefPicOrderCnt(colPic, refIdxCol, ListCol)) (8-12283)

tb = Clip3(-128, 127, PicOrderCnt(currPic) – RefPicOrderCnt(currPic, refIdxLX, LX)) (8-12324)

8.4.3.3.68.4.2.1.7 **Derivation process for chroma motion vectors**

[Ed.: (WJ) 4:2:0 assumption yet]

Inputs to this process are a luma motion vector mvLX and a reference index refIdLX.

Output of this process is a chroma motion vector mvCLX.

A chroma motion vector is derived from the corresponding luma motion vector.

For the derivation of the chroma motion vector mvCLX, the following applies.

mvCLX[0] = mvLX[0] (8-12485)

mvCLX[1] = mvLX[1] (8-12586)

8.4.2.2 Decoding process for inter prediction samples

Inputs to this process are:

- a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,
- a luma location (xB, yB) specifying the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current coding unit,
- a variable nCS specifying the size of the current coding unit,
- variables specifying the width and the height of the prediction unit, nPSW and nPSH,
- luma motion vectors mvL0 and mvL1, and chroma motion vectors mvCL0 and mvCL1,
- reference indices refIdxL0 and refIdxL1,
- prediction list utilization flags, predFlagL0 and predFlagL1.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000389

Outputs of this process are:

- a (nCS_L)x(nCS_L) array $predSamples_L$ of luma prediction samples, where nCS_L is derived as specified below,
- a (nCS_C)x(nCS_C) array $preSamples_{Cb}$ of chroma prediction samples for the component Cb, where nCS_C is derived as specified below,
- a (nCS_C)x(nCS_C) array $predSamples_{Cr}$ of chroma residual samples for the component Cr, where nCS_C is derived as specified below.

The variable nCS_L is set equal to nCS and the variable nCS_C is set equal to nCS >> 1. [Ed: (WJ) revisit for supporting other chroma formats]

Let $predSamplesL0_L$ and $predSamplesL1_L$ be (nPSW)x(nPSH) arrays of predicted luma sample values and $predSampleL0_{Cb}$, $predSampleL1_{Cb}$, $predSampleL0_{Cr}$, and $predSampleL1_{Cr}$ be (nPSW/2)x(nPSH/2) arrays of predicted chroma sample values.

For LX being replaced by either L0 or L1 in the variables predFlagLX, RefPicListX, refIdxLX, refPicLX, and predPartLX, the following is specified.

When predFlagLX is equal to 1, the following applies.

- The reference picture consisting of an ordered two-dimensional array $refPicLX_L$ of luma samples and two ordered two-dimensional arrays $refPicLX_{Cb}$ and $refPicLX_{Cr}$ of chroma samples is derived by invoking the process specified in subclause 8.4.2.2.1 with refIdxLX and RefPicListX given as input.
- The arrays $predSamplesLX_L$, $predSamplesLX_{Cb}$, and $predSamplesLX_{Cr}$ are derived by invoking the fractional sample interpolation process specified in subclause 8.4.2.2.2 with the luma locations (xC, yC), (xB, yB), the width an the height of the current prediction unit nPSW, nPSH, the motion vectors mvLX, mvCLX, and the reference arrays with $refPicLX_L$, $refPicLX_{Cb}$ and $refPicLX_{Cr}$ given as input.

The array $predSample_L$ of the prediction samples of luma component is derived by invoking the weighted sample prediction process specified in subclause 8.4.2.2.3 with the luma location (xB, yB), the width an the height of the current prediction unit nPSW, nPSH, and the sample arrays $predSamplesL0_L$ and $predSamplesL1_L$ as well as predFlagL0, predFlagL1 and $BitDepth_Y$ given as input.

For C being replaced by Cb, or Cr, the array $predSample_C$ of the prediction samples of component C is derived by invoking the weighted sample prediction process specified in subclause 8.4.2.2.3 with the chroma location (xB/2, yB/2), the width an the height of the current prediction unit $nPSW_C$ set equal to nPSW/2, $nPSH_C$ set equal to nPSH/2, and the sample arrays $predSamplesL0_C$ and $predSamplesL1_C$ as well as predFlagL0, predFlagL1 and $BitDepth_C$ given as input.

8.4.2.2.1 Reference picture selection process

[Ed: (WJ) same as AVC]

8.4.2.2.2 Fractional sample interpolation process

Inputs to this process are:

- a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,
- a luma location (xB, yB) specifying the top-left luma sample of the current prediction unit relative to the top left luma sample of the current coding unit,
- the width and height of this prediction unit, nPSW and nPSH, in luma-sample units,
- a luma motion vector mvLX given in quarter-luma-sample units,
- a chroma motion vector mvCLX given in eighth-chroma-sample units,
- the selected reference picture sample arrays $refPicLX_L$, $refPicLX_{Cb}$, and $refPicLX_{Cr}$.

Outputs of this process are:

- a (nPSW)x(nPSH) array $predSampleLX_L$ of prediction luma sample values,
- two (nPSW/2)x(nPSH/2) arrays $predSampleLX_{Cb}$, and $predSampleLX_{Cr}$ of prediction chroma sample values.

The location (xP, yP) given in full-sample units of the upper-left luma samples of the current prediction unit relative to the upper-left luma sample location of the given reference sample arrays is derived by

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000390

$xP = xC + xB$ (8-12687)
$yP = yC + yB$ (8-12788)

Let ($xInt_L$, $yInt_L$) be a luma location given in full-sample units and ($xFrac_L$, $yFrac_L$) be an offset given in quarter-sample units. These variables are used only inside this subclause for specifying general fractional-sample locations inside the reference sample arrays $refPicLX_L$, $refPicLX_{Cb}$, and $refPicLX_{Cr}$.

For each luma sample location ($0 <= x_L < nPSW, 0 <= y_L < nPSH$) inside the prediction luma sample array $predSampleLX_L$, the corresponding prediction luma sample value $predSampleLX_L[x_L, y_L]$ is derived as follows:

– The variables $xInt_L$, $yInt_L$, $xFrac_L$, and $yFrac_L$ are derived by

$xInt_L = xP + (mvLX[0] >> 2) + x_L$ (8-12889)
$yInt_L = yP + (mvLX[1] >> 2) + y_L$ (8-12990)

$xFrac_L = mvLX[0] \& 3$ (8-13094)
$yFrac_L = mvLX[1] \& 3$ (8-13195)

– The prediction luma sample value $predSampleLX_L[x_L, y_L]$ is derived by invoking the process specified in subclause 8.4.2.2.2.1 with ($xInt_L$, $yInt_L$), ($xFrac_L$, $yFrac_L$) and $refPicLX_L$ given as input.

Let ($xInt_C$, $yInt_C$) be a chroma location given in full-sample units and ($xFrac_C$, $yFrac_C$) be an offset given in one-eighth sample units. These variables are used only inside this subclause for specifying general fractional-sample locations inside the reference sample arrays $refPicLX_{Cb}$ and $refPicLX_{Cr}$.

For each chroma sample location ($0 <= x_C < nPSW/2, 0 <= y_C < nPSH/2$) inside the prediction chroma sample arrays $predSampleLX_{Cb}$ and $predSampleLX_{Cr}$, the corresponding prediction chroma sample values $predSampleLX_{Cb}[x_C, y_C]$ and $predSampleLX_{Cr}[x_C, y_C]$ are derived asfollows:

– The variables $xInt_C$, $yInt_C$, $xFrac_C$, and $yFrac_C$ are derived by

$xInt_C = (xP / 2) + (mvCLX[0] >> 3) + x_C$ (8-13293)
$yInt_C = (yP / 2) + (mvCLX[1] >> 3) + y_C$ (8-13394)

$xFrac_C = mvLX[0] \& 7$ (8-13495)
$yFrac_C = mvLX[1] \& 7$ (8-13596)

– The prediction sample value $predSampleLX_{Cb}[x_C, y_C]$ is derived by invoking the process specified in subclause 8.4.2.2.2.2 with ($xInt_C$, $yInt_C$), ($xFrac_C$, $yFrac_C$) and $refPicLX_{Cb}$ given as input.

– The prediction sample value $predSampleLX_{Cr}[x_C, y_C]$ is derived by invoking the process specified in subclause 8.4.2.2.2.2 with ($xInt_C$, $yInt_C$), ($xFrac_C$, $yFrac_C$) and $refPicLX_{Cr}$ given as input.

8.4.2.2.2.1 Luma sample interpolation process

Inputs to this process are:

– a luma location in full-sample units ($xInt_L$, $yInt_L$),

– a luma location in fractional-sample units ($xFrac_L$, $yFrac_L$),

– the luma reference sample array $refPicLX_L$.

Output of this process is a predicted luma sample value $predSampleLX_L[x_L, y_L]$

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000391

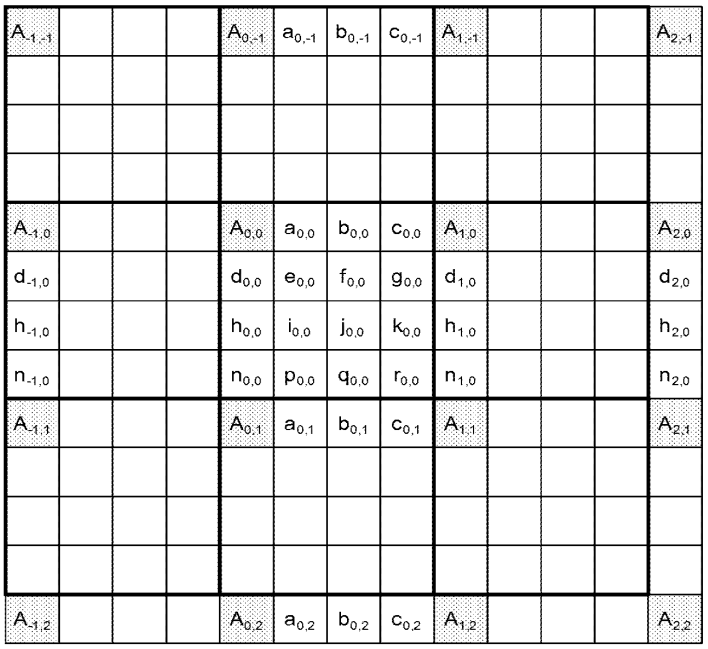


Figure 8-6 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation

In ~~Figure 8-6~~Figure 8-5, the positions labelled with upper-case letters $A_{i,j}$ within shaded blocks represent luma samples at full-sample locations inside the given two-dimensional array $refPicLX_L$ of luma samples. These samples may be used for generating the predicted luma sample value $predSampleLX_L[\, x_L, y_L \,]$. The locations $(xA_{i,j}, yA_{i,j})$ for each of the corresponding luma samples $A_{i,j}$ inside the given array $refPicLX_L$ of luma samples are derived as follows:

$$xA_{i,j} = Clip3(0, PicWidthInSamples_L - 1, xInt_L + i) \quad (8\text{-}13697)$$
$$yA_{i,j} = Clip3(0, PicHeightInSamples_L - 1, yInt_L + j) \quad (8\text{-}13798)$$

Variables shift1, shift2, shift3, offset1 and offset2 are derived as follows.

- The variable shift1 is set equal to $BitDepth_Y - 8$, the variable shift2 is set equal to $BitDepth_Y - 2$, and the variable shift3 is set equal to $14 - BitDepth_Y$.
- If the variable shift1 is equal to 0, the variable offset1 is set equal to 0, otherwise, the variable offset1 is set equal to $1 << (shift1 - 1)$.
- The variable offset2 is set equal to $1 << (shift2 - 1)$.

Given the luma samples $A_{i,j}$ at full-sample locations $(xA_{i,j}, yA_{i,j})$, the luma samples '$a_{0,0}$' to '$r_{0,0}$' at fractional sample positions are derived by the following rules.

- The samples labelled $a_{0,0}$, $b_{0,0}$, $c_{0,0}$, $d_{0,0}$, $h_{0,0}$, and $n_{0,0}$ shall be derived by applying the 8-tap filter to the nearest integer position samples and clipping the filtered value:

$$a_{0,0} = (-A_{-3,0} + 4*A_{-2,0} - 10*A_{-1,0} + 57*A_{0,0} + 19*A_{1,0} - 7*A_{2,0} + 3*A_{3,0} - A_{4,0} + offset1) >> shift1 \quad (8\text{-}13899)$$

$$b_{0,0} = (-A_{-3,0} + 4*A_{-2,0} - 11*A_{-1,0} + 40*A_{0,0} + 40*A_{1,0} - 11*A_{2,0} + 4*A_{3,0} - A_{4,0} + offset1) >> shift1 \quad (8\text{-}139100)$$

$$c_{0,0} = (-A_{-3,0} + 3*A_{-2,0} - 7*A_{-1,0} + 19*A_{0,0} + 57*A_{1,0} - 10*A_{2,0} + 4*A_{3,0} - A_{4,0} + offset1) >> shift1 \quad (8\text{-}140101)$$

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000392

$$d_{0,0} = (-A_{0,-3} + 4*A_{0,-2} - 10*A_{0,-1} + 57*A_{0,0} + 19*A_{0,1} - 7*A_{0,2} + 3*A_{0,3} - A_{0,4} + offset1) >> shift1 \quad (8\text{-}141)$$

$$h_{0,0} = (-A_{0,-3} + 4*A_{0,-2} - 11*A_{0,-1} + 40*A_{0,0} + 40*A_{0,1} - 11*A_{0,2} + 4*A_{0,3} - A_{0,4} + offset1) >> shift1 \quad (8\text{-}142)$$

$$n_{0,0} = (-A_{0,-3} + 3*A_{0,-2} - 7*A_{0,-1} + 19*A_{0,0} + 57*A_{0,1} - 10*A_{0,2} + 4*A_{0,3} - A_{0,4} + offset1) >> shift1 \quad (8\text{-}143)$$

– The samples labelled $e_{0,0}$, $f_{0,0}$, $g_{0,0}$, $i_{0,0}$, $j_{0,0}$, $k_{0,0}$, $p_{0,0}$, $q_{0,0}$ and $r_{0,0}$ shall be derived by first calculating intermediate values denoted as $d1_{i,0}$, $h1_{i,0}$ and $n1_{i,0}$ where i = -3..4 by applying the 8-tap filter to the nearest integer position samples in vertical direction: [Ed: (WJ) horizontal first yields the same result]

$$d1_{i,0} = -A_{i,-3} + 4*A_{i,-2} - 10*A_{i,-1} + 57*A_{i,0} + 19*A_{i,1} - 7*A_{i,2} + 3*A_{i,3} - A_{i,4} \quad (8\text{-}144)$$

$$h1_{i,0} = -A_{i,-3} + 4*A_{i,-2} - 11*A_{i,-1} + 40*A_{i,0} + 40*A_{i,1} - 11*A_{i,2} + 4*A_{i,3} - A_{i,4} \quad (8\text{-}145)$$

$$n1_{i,0} = -A_{i,-3} + 3*A_{i,-2} - 7*A_{i,-1} + 19*A_{i,0} + 57*A_{i,1} - 10*A_{i,2} + 4*A_{i,3} - A_{i,4} \quad (8\text{-}146)$$

– The final prediction values $e_{0,0}$, $f_{0,0}$, $g_{0,0}$, $i_{0,0}$, $j_{0,0}$, $k_{0,0}$, $p_{0,0}$, $q_{0,0}$ and $r_{0,0}$ shall be derived by applying the 8-tap filter to the intermediate values $d1_{i,0}$, $h1_{i,0}$ and $n1_{i,0}$ where i = -3..4 in horizontal direction:

$$e_{0,0} = (-d1_{-3,0} + 4*d1_{-2,0} - 10*d1_{-1,0} + 57*d1_{0,0} + 19*d1_{1,0} - 7*d1_{2,0} + 3*d1_{3,0} - d1_{4,0} + offset2) >> shift2 \quad (8\text{-}147)$$

$$f_{0,0} = (-d1_{-3,0} + 4*d1_{-2,0} - 11*d1_{-1,0} + 40*d1_{0,0} + 40*d1_{1,0} - 11*d1_{2,0} + 4*d1_{3,0} - d1_{4,0} + offset2) >> shift2 \quad (8\text{-}148)$$

$$g_{0,0} = (-d1_{-3,0} + 3*d1_{-2,0} - 7*d1_{-1,0} + 19*d1_{0,0} + 57*d1_{1,0} - 10*d1_{2,0} + 4*d1_{3,0} - d1_{4,0} + offset2) >> shift2 \quad (8\text{-}149)$$

$$i_{0,0} = (-h1_{-3,0} + 4*h1_{-2,0} - 10*h1_{-1,0} + 57*h1_{0,0} + 19*h1_{1,0} - 7*h1_{2,0} + 3*h1_{3,0} - h1_{4,0} + offset2) >> shift2 \quad (8\text{-}150)$$

$$j_{0,0} = (-h1_{-3,0} + 4*h1_{-2,0} - 11*h1_{-1,0} + 40*h1_{0,0} + 40*h1_{1,0} - 11*h1_{2,0} + 4*h1_{3,0} - h1_{4,0} + offset2) >> shift2 \quad (8\text{-}151)$$

$$k_{0,0} = (-h1_{-3,0} + 3*h1_{-2,0} - 7*h1_{-1,0} + 19*h1_{0,0} + 57*h1_{1,0} - 10*h1_{2,0} + 4*h1_{3,0} - h1_{4,0} + offset2) >> shift2 \quad (8\text{-}152)$$

$$p_{0,0} = (-n1_{-3,0} + 4*n1_{-2,0} - 10*n1_{-1,0} + 57*n1_{0,0} + 19*n1_{1,0} - 7*n1_{2,0} + 3*n1_{3,0} - n1_{4,0} + offset2) >> shift2 \quad (8\text{-}153)$$

$$q_{0,0} = (-n1_{-3,0} + 4*n1_{-2,0} - 11*n1_{-1,0} + 40*n1_{0,0} + 40*n1_{1,0} - 11*n1_{2,0} + 4*n1_{3,0} - n1_{4,0} + offset2) >> shift2 \quad (8\text{-}154)$$

$$r_{0,0} = (-n1_{-3,0} + 3*n1_{-2,0} - 7*n1_{-1,0} + 19*n1_{0,0} + 57*n1_{1,0} - 10*n1_{2,0} + 4*n1_{3,0} - n1_{4,0} + offset2) >> shift2 \quad (8\text{-}155)$$

The positions labelled with lower-case letters within un-shaded blocks represent luma samples at quarter-pel sample fractional locations. The luma location offset in fractional-sample units ($xFrac_L$, $yFrac_L$) specifies which of the generated luma samples at full-sample and fractional-sample locations is assigned to the predicted luma sample value $predSampleLX_L[x_L, y_L]$. This assignment is done according to Table 8-11~~Table 8-9~~. The value of $predSampleLX_L[x_L, y_L]$ shall be the output.

Table 8-11~~8-9~~ – Assignment of the luma prediction sample $predSampleLX_L[x_L, y_L]$

xFracL	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
yFracL	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$predSampleLX_L[x_L, y_L]$	A << shift3	d	h	n	a	e	i	p	b	f	j	q	c	g	k	r

8.4.2.2.2.2 Chroma sample interpolation process

Inputs to this process are:

– a chroma location in full-sample units ($xInt_C$, $yInt_C$),

– a chroma location in fractional-sample units ($xFrac_C$, $yFrac_C$),

– the chroma reference sample array $refPicLX_C$.

Output of this process is a predicted chroma sample value $predSampleLX_C[x_C, y_C]$

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000393

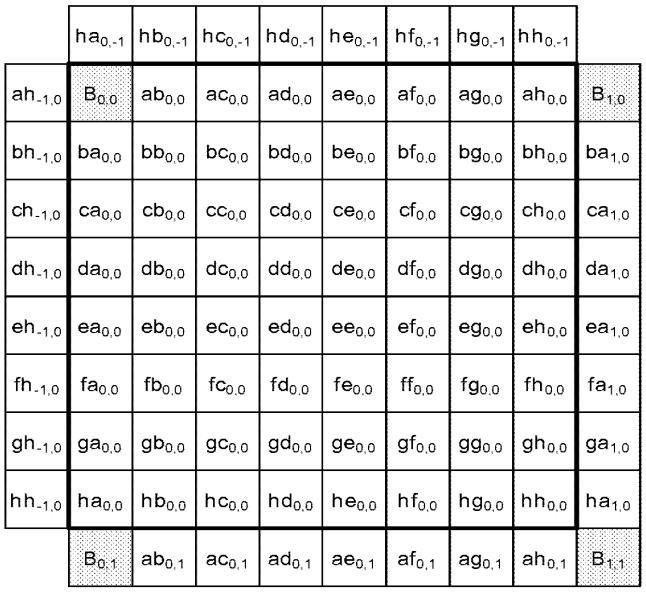


Figure 8-7 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for eighth sample chroma interpolation

In ~~Figure 8-6~~Figure 8-5, the positions labelled with upper-case letters $B_{i,j}$ within shaded blocks represent chroma samples at full-sample locations inside the given two-dimensional array $refPicLX_C$ of chroma samples. These samples may be used for generating the predicted chroma sample value $predSampleLX_C[\, x_C, y_C \,]$. The locations $(xB_{i,j}, yB_{i,j})$ for each of the corresponding chroma samples $B_{i,j}$ inside the given array $refPicLX_C$ of chroma samples are derived as follows:

$$xB_{i,j} = Clip3(0, PicWidthInSamples_C - 1, xInt_C + i) \quad (8\text{-}~~156~~117)$$
$$yB_{i,j} = Clip3(0, PicHeightInSamples_C - 1, yInt_C + j) \quad (8\text{-}~~157~~118)$$

Variables shift1, shift2, shift3, offset1 and offset2 are derived as follows.

- The variable shift1 is set equal to $BitDepth_C - 8$, the variable shift2 is set equal to $BitDepth_C - 2$, and the variable shift3 is set equal to $14 - BitDepth_C$.
- If the variable shift1 is equal to 0, the variable offset1 is set equal to 0, otherwise, the variable offset1 is set equal to $1 << (shift1 - 1)$.
- The variable offset2 is set equal to $1 << (shift2 - 1)$.

Given the chroma samples $B_{i,j}$ at full-sample locations $(xB_{i,j}, yB_{i,j})$, the chroma samples '$ab_{0,0}$' to '$hh_{0,0}$' at fractional sample positions are derived by the following rules.

- The samples labelled $ab_{0,0}$, $ac_{0,0}$, $ad_{0,0}$, $ae_{0,0}$, $af_{0,0}$, $ag_{0,0}$, and $ah_{0,0}$ shall be derived by applying the 4-tap filter to the nearest integer position samples and clipping the filtered value:

$$ab_{0,0} = (-3*B_{-1,0} + 60*B_{0,0} + 8*B_{1,0} - B_{2,0} + offset1) >> shift1 \quad (8\text{-}~~158~~119)$$

$$ac_{0,0} = (-4*B_{-1,0} + 54*B_{0,0} + 16*B_{1,0} - 2*B_{2,0} + offset1) >> shift1 \quad (8\text{-}~~159~~120)$$

$$ad_{0,0} = (-5*B_{-1,0} + 46*B_{0,0} + 27*B_{1,0} - 4*B_{2,0} + offset1) >> shift1 \quad (8\text{-}~~160~~121)$$

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000394

$ae_{0,0} = (-4*B_{-1,0} + 36*B_{0,0} + 36*B_{1,0} - 4*B_{2,0} + offset1) >> shift1$ (8-161)

$af_{0,0} = (-4*B_{-1,0} + 27*B_{0,0} + 46*B_{1,0} - 5*B_{2,0} + offset1) >> shift1$ (8-162)

$ag_{0,0} = (-2*B_{-1,0} + 16*B_{0,0} + 54*B_{1,0} - 4*B_{2,0} + offset1) >> shift1$ (8-163)

$ah_{0,0} = (-B_{-1,0} + 8*B_{0,0} + 60*B_{1,0} - 3*B_{2,0} + offset1) >> shift1$ (8-164)

– The samples labelled $ba_{0,0}$, $ca_{0,0}$, $da_{0,0}$, $ea_{0,0}$, $fa_{0,0}$, $ga_{0,0}$, and $ha_{0,0}$ shall be derived by applying the 4-tap filter to the nearest integer position samples and clipping the filtered value:

$ba_{0,0} = (-3*B_{0,-1} + 60*B_{0,0} + 8*B_{0,1} - B_{0,2} + offset1) >> shift1$ (8-165)

$ca_{0,0} = (-4*B_{0,-1} + 54*B_{0,0} + 16*B_{0,1} - 2*B_{0,2} + offset1) >> shift1$ (8-166)

$da_{0,0} = (-5*B_{0,-1} + 46*B_{0,0} + 27*B_{0,1} - 4*B_{0,2} + offset1) >> shift1$ (8-167)

$ea_{0,0} = (-4*B_{0,-1} + 36*B_{0,0} + 36*B_{0,1} - 4*B_{0,2} + offset1) >> shift1$ (8-168)

$fa_{0,0} = (-4*B_{0,-1} + 27*B_{0,0} + 46*B_{0,1} - 5*B_{0,2} + offset1) >> shift1$ (8-169)

$ga_{0,0} = (-2*B_{0,-1} + 16*B_{0,0} + 54*B_{0,1} - 4*B_{0,2} + offset1) >> shift1$ (8-170)

$ha_{0,0} = (-B_{0,-1} + 8*B_{0,0} + 60*B_{0,1} - 3*B_{0,2} + offset1) >> shift1$ (8-171)

– The samples labelled $Xb_{0,0}$, $Xc_{0,0}$, $Xd_{0,0}$, $Xe_{0,0}$, $Xf_{0,0}$, $Xg_{0,0}$ and $Xh_{0,0}$ for X being replaced by b, c, d, e, f, g and h, respectively, shall be derived by first calculating intermediate values denoted as $ba_{i,0}$, $ca_{i,0}$, $da_{i,0}$, $ea_{i,0}$, $fa_{i,0}$, $ga_{i,0}$ and $ha_{i,0}$ where i = -1..2 by applying the 4-tap filter to the nearest integer position samples in vertical direction: [Ed: (WJ) horizontal first yields the same result]

$ba_{i,0} = -3*B_{0,-1} + 60*B_{0,0} + 8*B_{0,1} - B_{0,2}$ (8-172)

$ca_{i,0} = -4*B_{0,-1} + 54*B_{0,0} + 16*B_{0,1} - 2*B_{0,2}$ (8-173)

$da_{i,0} = -5*B_{0,-1} + 46*B_{0,0} + 27*B_{0,1} - 4*B_{0,2}$ (8-174)

$ea_{i,0} = -4*B_{0,-1} + 36*B_{0,0} + 36*B_{0,1} - 4*B_{0,2}$ (8-175)

$fa_{i,0} = -4*B_{0,-1} + 27*B_{0,0} + 46*B_{0,1} - 5*B_{0,2}$ (8-176)

$ga_{i,0} = -2*B_{0,-1} + 16*B_{0,0} + 54*B_{0,1} - 4*B_{0,2}$ (8-177)

$ha_{i,0} = -B_{0,-1} + 8*B_{0,0} + 60*B_{0,1} - 3*B_{0,2}$ (8-178)

– The final prediction values $Xb_{0,0}$, $Xc_{0,0}$, $Xd_{0,0}$, $Xe_{0,0}$, $Xf_{0,0}$, $Xg_{0,0}$ and $Xh_{0,0}$ for X being replaced by b, c, d, e, f, g and h, respectively, shall be derived by applying the 4-tap filter to the intermediate values $Xa_{i,0}$ where i = -1..2 in horizontal direction:

$Xb_{0,0} = (-3*Xa_{-1,0} + 60*Xa_{0,0} + 8*Xa_{1,0} - Xa_{2,0} + offset2) >> shift2$ (8-179)

$Xc_{0,0} = (-4*Xa_{-1,0} + 54*Xa_{0,0} + 16*Xa_{1,0} - 2*Xa_{2,0} + offset2) >> shift2$ (8-180)

$Xd_{0,0} = (-5*Xa_{-1,0} + 46*Xa_{0,0} + 27*Xa_{1,0} - 4*Xa_{2,0} + offset2) >> shift2$ (8-181)

$Xe_{0,0} = (-4*Xa_{-1,0} + 36*Xa_{0,0} + 36*Xa_{1,0} - 4*Xa_{2,0} + offset2) >> shift2$ (8-182)

$Xf_{0,0} = (-4*Xa_{-1,0} + 27*Xa_{0,0} + 46*Xa_{1,0} - 5*Xa_{2,0} + offset2) >> shift2$ (8-183)

$Xg_{0,0} = (-2*Xa_{-1,0} + 16*Xa_{0,0} + 54*Xa_{1,0} - 4*Xa_{2,0} + offset2) >> shift2$ (8-184)

$Xh_{0,0} = (-Xa_{-1,0} + 8*Xa_{0,0} + 60*Xa_{1,0} - 3*Xa_{2,0} + offset2) >> shift2$ (8-185)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000395

The positions labelled with lower-case letters within un-shaded blocks represent chroma samples at eighth-pel sample fractional locations. The chroma location offset in fractional-sample units ($xFrac_C$, $yFrac_C$) specifies which of the generated chroma samples at full-sample and fractional-sample locations is assigned to the predicted chroma sample value $predSampleLX_C[x_C, y_C]$. This assignment is done according to Table 8-12~~Table 8-10~~. The value of $predSampleLX_C[x_C, y_C]$ shall be the output.

Table 8-12~~8-10~~ – Assignment of the chroma prediction sample $predSampleLX_C[x_C, y_C]$ for (X, Y) being replaced by (1, b), (2, c), (3, d), (4, e), (5, f), (6, g), and (7, h), respectively

xFracC	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
yFracC	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
$predSampleLX_C[x_C, y_C]$	B << shift3	ba	ca	da	ea	fa	ga	ha	aY	bY	cY	dY	eY	fY	gY	hY

8.4.2.2.3 Weighted sample prediction process

[Ed.: (KU) only the default weighted prediction part is added for bi-directional averaging]

Inputs to this process are:

- a location (xB, yB) specifying the top-left sample of the current prediction unit relative to the top left sample of the current coding unit,
- the width and height of this prediction unit, nPSW and nPSH,
- two (nPSW)x(nPSH) arrays predSamplesL0 and predSamplesL1,
- prediction list utilization flags, predFlagL0 and predFlagL1,
- the bit-depth of the chroma component, bitDepth.

Outputs of this process are:

- the (nPSW)x(nPSH) array predSamples of prediction sample values.

Variables shift1, shift2, offset1 and offset2 are derived as follows.

- The variable shift1 is set equal to 14 – bitDepth and the variable shift2 is set equal to 15 – bitDepth,
- The variable offset1 is set equal to 1 << (shift1 – 1) and the variable offset2 is set equal to 1 << (shift2 – 1).

Depending on the value of predFlagL0 and predFlagL1, the prediction samples predSamples[x, y] with x = 0..(nPSW)-1 and y = 0..(nPSH)-1 are derived as follows.

- If predFlagL0 is equal to 1 and predFlagL1 is equal to 0,

$$predSamples[x, y] = (predSamplesL0[x, y] + offset1) >> shift1 \quad (8\text{-}186\text{~~147~~})$$

- Otherwise, if predFlagL0 is equal to 0 and predFlagL1 is equal to 1,

$$predSamples[x, y] = (predSamplesL1[x, y] + offset1) >> shift1 \quad (8\text{-}187\text{~~148~~})$$

- Otherwise (both predFlagL0 and predFlagL1 are equal to 1),

$$predSamples[x, y] = Clip3(0, (1 << bitDepth) – 1 , (predSamplesL0[x, y] + predSamplesL1[x, y] + offset2) >> shift2) \quad (8\text{-}188\text{~~149~~})$$

8.4.3 Decoding process for the residual signal of coding units coded in inter prediction mode

Inputs to this process are:

- a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top-left luma sample of the current picture,
- a variable log2CUSize specifying the size of the current coding unit.

Outputs of this process are:

- a $(nCS_L)x(nCS_L)$ array $resSamples_L$ of luma residual samples, where nCS_L is derived as specified below,

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000396

– a (nCS_C)x(nCS_C) array $resSamples_{Cb}$ of chroma residual samples for the component Cb, where nCS_C is derived as specified below,

– a (nCS_C)x(nCS_C) array $resSamples_{Cr}$ of chroma residual samples for the component Cr, where nCS_C is derived as specified below.

The variable nCS_L is set equal to 1 << log2CUSize and the variable nCS_C is set equal to (1 << log2CUSize) >> 1.

Let $resSamples_L$ be a (nCS_L)x(nCS_L) array of luma residual samples and let $resSamples_{Cb}$ and $resSamples_{Cr}$ be two (nCS_C)x(nCS_C) arrays of chroma residual samples.

Depending on no_residual_data_flag, the following applies:

– If no_residual_data_flag is equal to 1, all samples of the (nCS_L)x(nCS_L) array $resSamples_L$ and all samples of the two (nCS_C)x(nCS_C) arrays $resSamples_{Cb}$ and $resSamples_{Cr}$ are set equal to 0.

– Otherwise (no_residual_data_flag is equal to 0), the following ordered steps apply:

1. The decoding process for luma residual blocks as specified in subclause 8.4.3.1 below is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to (0, 0), the variable log2TrafoSize set equal to log2CUSize, the variable trafoDepth set equal to 0, the variable nCS set equal to nCSL, and the (nCSL)x(nCSL) array resSamplesL as the inputs and the output is a modified version of the (nCSL)x(nCSL) array resSamplesL.

2. The decoding process for chroma residual blocks as specified in subclause 8.4.3.2 below is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to (0, 0), the variable log2TrafoSize set equal to log2CUSize, the variable trafoDepth set equal to 0, the variable cIdx set equal to 1, the variable nCS set equal to nCS_C, and the (nCS_C)x(nCS_C) array $resSamples_{Cb}$ as the inputs and the output is a modified version of the (nCS_C)x(nCS_C) array $resSamples_{Cb}$.

3. The decoding process for chroma residual blocks as specified in subclause 8.4.3.2 below is invoked with the luma location (xC, yC), the luma location (xB, yB) set equal to (0, 0), the variable log2TrafoSize set equal to log2CUSize, the variable trafoDepth set equal to 0, the variable cIdx set equal to 2, the variable nCS set equal to nCS_C, and the (nCS_C)x(nCS_C) array $resSamples_{Cr}$ as the inputs and the output is a modified version of the (nCS_C)x(nCS_C) array $resSamples_{Cr}$.

8.4.3.1 Decoding process for luma residual blocks

Inputs to this process are:

– a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top-left luma sample of the current picture,

– a luma location (xB, yB) specifying the top-left luma sample of the current block relative to the top-left luma sample of the current coding unit,

– a variable log2TrafoSize specifying the size of the current block,

– a variable trafoDepth specifying the hierarchy depth of the current block relative to the coding unit,

– a variable nCS specifying the size, in luma samples, of the current coding unit,

– a (nCS)x(nCS) array resSamples of luma residual samples.

Output of this process is:

– a modified version of the (nCS)x(nCS) array of luma residual samples.

Depending split_transform_flag[xB][yB][trafoDepth], the following applies:

– If split_transform_flag[xB][yB][trafoDepth] is equal to 1, the following ordered steps apply:

1. The variable xB1 is set equal to xB + ((1 << log2TrafoSize) >> 1).

2. The variable yB1 is set equal to yB + ((1 << log2TrafoSize) >> 1).

3. The decoding process for luma residual blocks as specified in this subclause is invoked with the luma location (xC, yC), the luma location (xB, yB), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the variable nCS, and the (nCS)x(nCS) array resSamples as the inputs and the output is a modified version of the (nCS)x(nCS) array resSamples.

4. The decoding process for luma residual blocks as specified in this subclause is invoked with the luma location (xC, yC), the luma location (xB1, yB), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000397

trafoDepth set equal to trafoDepth + 1, the variable nCS, and the (nCS)x(nCS) array resSamples as the inputs and the output is a modified version of the (nCS)x(nCS) array resSamples.

5. The decoding process for luma residual blocks as specified in this subclause is invoked with the luma location (xC, yC), the luma location (xB, yB1), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the variable nCS, and the (nCS)x(nCS) array resSamples as the inputs and the output is a modified version of the (nCS)x(nCS) array resSamples.

6. The decoding process for luma residual blocks as specified in this subclause is invoked with the luma location (xC, yC), the luma location (xB1, yB1), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the variable nCS, and the (nCS)x(nCS) array resSamples as the inputs and the output is a modified version of the (nCS)x(nCS) array resSamples.

– Otherwise (split_transform_flag[xB][yB][trafoDepth] is equal to 0), the following ordered steps apply:

1. The variable nS is set equal to 1 << log2TrafoSize.

2. The scaling and transformation process as specified in subclause 8.5.18.5.1 is invoked with the luma location (xC + xB, yC +yB), the variable trafoDepth, the variable cIdx set equal to 0, and the transform size trafoSize set equal to nS as the inputs and the output is a (nS)x(nS) array resSamplesBlock.

3. The array construction process as specified in subclause 8.5.5 is invoked with the luma location (xB, yB), the variable cIdx set equal to 0, the variable inputArraySize set equal to nS, the variable outputArraySize set equal to nCS, the (nS)x(nS) array resSamplesBlock, and the (nCS)x(nCS) array resSamples as the inputs and the output is a modified version of the (nCS)x(nCS) array resSamples.

8.4.3.2 Decoding process for chroma residual blocks

Inputs to this process are:

– a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top-left luma sample of the current picture,

– a luma location (xB, yB) specifying the top-left luma sample of the current block relative to the top-left luma sample of the current coding unit,

– a variable log2TrafoSize specifying the size of the current block,

– a variable trafoDepth specifying the hierarchy depth of the current block relative to the coding unit,

– a variable cIdx specifying the chroma component of the current block,

– a variable nCS specifying the size, in chroma samples, of the current coding unit,

– a (nCS)x(nCS) array resSamples of chroma residual samples.

Output of this process is:

– a modified version of the (nCS)x(nCS) array of chroma residual samples.

The variable splitChromaFlag is derived as follows:

– If split_transform_flag[xB][yB][trafoDepth] is equal to 1 and log2TrafoSize is greater than Log2MinTrafoSize + 1, splitChromaFlag is set equal to 1.

– Otherwise (split_transform_flag[xB][yB][trafoDepth] is equal to 0 or log2TrafoSize is equal to Log2MinTrafoSize + 1), splitChromaFlag is set equal to 0.

Depending splitChromaFlag, the following applies:

– If splitChromaFlag is equal to 1, the following ordered steps apply:

1. The variable xB1 is set equal to xB + ((1 << log2TrafoSize) >> 1).

2. The variable yB1 is set equal to yB + ((1 << log2TrafoSize) >> 1).

3. The decoding process for residual chroma blocks as specified in this subclause is invoked with the luma location (xC, yC), the luma location (xB, yB), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the variable cIdx, the variable nCS, and the (nCS)x(nCS) array resSamples as the inputs and the output is a modified version of the (nCS)x(nCS) array resSamples.

4. The decoding process for residual chroma blocks as specified in this subclause is invoked with the luma location (xC, yC), the luma location (xB1, yB), the variable log2TrafoSize set equal to log2TrafoSize − 1, the

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000398

variable trafoDepth set equal to trafoDepth + 1, the variable cIdx, the variable nCS, and the (nCS)x(nCS) array resSamples as the inputs and the output is a modified version of the (nCS)x(nCS) array resSamples.

5. The decoding process for residual chroma blocks as specified in this subclause is invoked with the luma location (xC, yC), the luma location (xB, yB1), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the variable cIdx, the variable nCS, and the (nCS)x(nCS) array resSamples as the inputs and the output is a modified version of the (nCS)x(nCS) array resSamples.

6. The decoding process for residual chroma blocks as specified in this subclause is invoked with the luma location (xC, yC), the luma location (xB1, yB1), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the variable cIdx, the variable nCS, and the (nCS)x(nCS) array resSamples as the inputs and the output is a modified version of the (nCS)x(nCS) array resSamples.

– Otherwise (splitChromaFlag is equal to 0), the following ordered steps apply:

1. The variable nS is set equal to (1 << log2TrafoSize) >> 1.

2. The scaling and transformation process as specified in subclause 8.5.1 is invoked with the luma location (xC + xB, yC +yB), the variable trafoDepth, the variable cIdx, and the transform size trafoSize set equal to nS as the inputs and the output is a (nS)x(nS) array resSamplesBlock.

3. The array construction process as specified in subclause 8.5.5 is invoked with the luma location (xB, yB), the variable cIdx, the variable inputArraySize set equal to nS, the variable outputArraySize set equal to nCS, the (nS)x(nS) array resSamplesBlock, and the (nCS)x(nCS) array resSamples as the inputs and the output is a modified version of the (nCS)x(nCS) array resSamples.

8.5 Scaling, transformation and array construction process prior to deblocking filter process

8.5.1 Scaling and transformation process

Inputs to this process are:

– a luma location (xT, yT) specifying the top-left luma sample of the current transform unit relative to the top-left luma sample of the current picture,

– a variable trafoDepth specifying the hierarchy depth of the current block relative to the coding unit,

– a variable cIdx specifying the chroma component of the current block,

– a variable nS specifying the size of the current transform unit.

Output of this process is:

– a modified version of the (nS)x(nS) array of residual samples r with elements r_{ij}.

The quantization parameter qP is derived as follows.

– If cIdx is equal to 0,

$$qP = QP'_Y \qquad (8\text{-}189\text{-}150)$$

– Otherwise,

$$qP = QP'_C \qquad (8\text{-}190\text{-}151)$$

The (nS)x(nS) array of residual samples r are derived as specified in the following ordered steps.

1. The inverse scanning process for transform coefficients as specified in subclause 8.5.2 is invoked with the size of the current transform unit nS, the list of transform coefficients transCoeffLevel[xT][yT][trafoDepth][cIdx] and the variable cIdx as the inputs and the output is a (nS)x(nS) array c_{ij}.

2. The scaling process for transform coefficients as specified in subclause 8.5.3 is invoked with the size of the transform unit nS, the (nS)x(nS) array c, the chroma component variable cIdx and the quantization parameter qP as the inputs and the output is a scaled transform coefficient (nS)x(nS) array d.

3. The transformation process for scaled transform coefficients as specified in subclause 8.5.4 is invoked with the size of the transform unit nS, the scaled transform coefficient (nS)x(nS) array d and the chroma component variable cIdx as the inputs and the output is a residual samples (nS)x(nS) array r.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000399

8.5.2 Inverse scanning process for transform coefficients

Inputs to this process are:

- a variable nS specifying the size of the current transform unit,
- a list of (nS)x(nS) values transCoeffLevel[xT][yT][trafoDepth][cIdx].
- a variable cIdx specifying the chroma component of the current block,

Output of this process is a variable c containing a two-dimensional array of (nS)x(nS) values.

- The variable c[i][j] which is located in the (i, j) position in the array c is derived as follows.
- The current transform unit is divided into (nS >> 2) * (nS >> 2) subsets of 4 * 4 samples. The subsets are processed in zig-zag scan order. The position (k0, l0) in the array c, corresponding to the top left position of a subset s, is derived as follows.

```
k0 = ZigZag[ log2( nS >> 2 ) ][ s ][ 0 ] << 2
l0 = ZigZag[ log2( nS >> 2 ) ][ s ][ 1 ] << 2,
                                    with s = 0.. ( ( nS >> 2 ) * ( nS >> 2 ) ) – 1 )
                                                                    (8-191452)
```

  - For every subset s, the 16 transform coefficient level at scanning position n are mapped to the position (i, j) in the array c using the reverse zig-zag scan.

```
i = k0 + ZigZag[ 2 ][ n ][ 0 ]
j = l0 + ZigZag[ 2 ][ n ][ 1 ]
c[ i ][ j ] = transCoeffLevel[ xT ][ yT ][ trafoDepth ][ cIdx ][ n + ( s << 4 ) ],
                                                    with n = 15..0
                                                                    (8-192452)
```

8.5.3 Scaling process for transform coefficients

Inputs of this process are:

- a variable nS specifying the size of the current transform unit,
- a (nS)x(nS) array c of transform coefficients with elements c_{ij},
- a variable cIdx specifying the chroma component of the current block,
- a variable qP specifying the quantization parameter.

Output of this process is scaled transform coefficients as a (nS)x(nS) array of d with elements d_{ij}.

The variable trafoPrecisionExt is derived as follows:

- If nS is greater than 4,
  - If cIdx is equal to 0 and bit_depth_luma_minus8 is equal to 0, trafoPrecisionExt is set equal to 2.
  - Otherwise, if cIdx is not equal to 0 and bit_depth_chroma_minus8 is equal to 0, trafoPrecisionExt is set equal to 2.
  - Otherwise, trafoPrecisionExt is set equal to 0.
- Otherwise, trafoPrecisionExt is set equal to 0.

[Ed.: (WJ) wait decision on transform precision extension]

The scaled transform coefficient array dij is derived as follows.

$$d_{ij} = (c_{ij} * LevelScale_{(nS)x(nS)}[qP\%6][i][j]) << (qP/6 + trafoPrecisionExt), \text{ with } i, j = 0..nS-1 \quad (8\text{-}193452)$$

[Ed: (WJ) $LevelScale_{(nS)x(nS)}$ should be very large table up to 32x32. Recent contributions (JCTVC-C209 and JCTVC-C255) shown that this matrix could be replaced by one constants per qP without coding efficiency penalty. Maybe it is better to wait.]

[Ed.: (WJ) dependent on transformation process – should be modified again after transformation process is fixed]

8.5.4 Transformation process for scaled transform coefficients

Inputs of this process are:

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000400

– a variable nS specifying the size of the current transform unit,

– a (nS)x(nS) array d of scaled transform coefficients with elements d_{ij}.

– a variable cIdx specifying the chroma component of the current block,

Output of this process is residual samples as a (nS)x(nS) array r with elements r_{ij}.

The variable trafoPrecisionExt is derived as follows:

– If nS is greater than 4,

  – If cIdx is equal to 0 and bit_depth_luma_minus8 is equal to 0, trafoPrecisionExt is set equal to 4.

  – Otherwise, if cIdx is not equal to 0 and bit_depth_chroma_minus8 is equal to 0, trafoPrecisionExt is set equal to 4.

  – Otherwise, trafoPrecisionExt is set equal to 0.

– Otherwise, trafoPrecisionExt is set equal to 0.

[Ed.: (WJ) wait decision on transform precision extension]

[Ed: (WJ) derivation of trafoPrecisionExt is duplicated to that of the scaling process. Better to remove one of two.]

~~Depending on cIdx, a variable planarFlag is derived as follows:~~

~~If cIdx is equal to 0, planarFlag is set equal to planar_flag_luma~~

~~Otherwise, planarFlag is set equal to planar_flag_chroma (Ed. (WJ): it is not necessary, since DST is not applied to chroma anyway)~~

~~Depending on PredMode and IntraPredMode, the following applies:~~

~~If PredMode is equal to MODE_INTRA, nS is equal to 4, and cIdx is equal to 0, the variables horizTrType and vertTrType are specified as Table 8-13Table 8-11 with IntraPredMode as input. [Ed. (WJ): DST is applied only for luma 4x4 block]~~

~~Otherwise, the variables horizTrType and vertTrType are set equal to 0.~~

~~Table 8-138-11 – Specification of horizTrType and vertTrType~~

~~IntraPredMode~~	~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	~~11~~	~~12~~	~~13~~	~~14~~	~~15~~	~~16~~	~~17~~
~~horizTrType~~	~~0~~	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~	~~1~~
~~vertTrType~~	~~1~~	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~

~~IntraPredMode~~	~~18~~	~~19~~	~~20~~	~~21~~	~~22~~	~~23~~	~~24~~	~~25~~	~~26~~	~~27~~	~~28~~	~~29~~	~~30~~	~~31~~	~~32~~	~~33~~	~~34~~
~~horizTrType~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~
~~vertTrType~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~

The constructed residual samples are derived as specified in the following ordered steps.

1. Each (horizontal) row of scaled transform coefficients d_{ij} (i, j=0..nS-1) is transformed to e_{ij} (i, j=0..nS-1) by invoking the one-dimensional transformation process as specified in subclause 8.5.4.1 to ~~8.5.4.4~~8.5.4.4 according to the size of the transform unit nS, with the (nS)x(nS) array d ~~and the transform type variable horizTrType~~ as the input~~,~~ and the output is the (nS)x(nS) array e.

2. Each (vertical) column of the resulting matrix e_{ij} (i, j=0..nS-1) is transformed to f_{ij} (i, j=0..nS-1) by invoking the one-dimensional transformation process as specified in subclause 8.5.4.1 to 8.5.4.4 according to the size of the transform unit nS, with the (nS)x(nS) array e ~~and the transform type variable vertTrType~~ as the input~~,~~ and the output is the (nS)x(nS) array f.

3. If nS is equal to 4 or 8, the residual sample value r_{ij} is derived by

$$r_{ij} = (f_{ij} + 2^{5+trafoPrecisionExt}) >> (6 + trafoPrecisionExt), \text{ with } i, j = 0..(nS)x(nS)-1 \qquad (8\text{-}~~194~~159)$$

서식 있음: 들여쓰기: 왼쪽: 0 cm, 내어쓰기: 4.26 글자, 탭: 2.13 글자, 왼쪽

서식 있음: 캡션

서식 있음: 들여쓰기: 왼쪽: 0 cm, 내어쓰기: 4.26 글자, 탭: 2.13 글자, 왼쪽

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000401

4. Otherwise (nS is equal to 16 or 32), the residual sample value r_{ij} is derived by

$$r_{ij} = (f_{ij} + 2^{9+trafoPrecisionExt}) >> (10 + trafoPrecisionExt), \text{ with } i, j=0.. (nS)x(nS)-1 \quad (8-195)$$

[Ed.: (WJ) dependent on transformation process – should be modified again after transformation process is fixed]

8.5.4.1 Transformation process for 4 samples

Inputs of this process are an array of 16 samples of the scaled transform coefficients x with elements x_i, with i = 0..3 and the transform type variable trType.

Output of this process is an array of 16 samples of the residual samples y with elements y_i, with i = 0..3.

[Ed.: (WJ) TBD] Depending on trType, the following applies:

If trType is equal to 1, the following ordered steps apply:

1. A set of intermediate values s_0, s_1, s_2 and s_3 is calculated as follows:

$$s_0 = x_0 + x_2 \quad (8-196)$$
$$s_1 = x_2 + x_3 \quad (8-197)$$
$$s_2 = x_0 - x_3 \quad (8-198)$$
$$s_3 = 74 * x_1 \quad (8-199)$$

2. The output values y_i with i = 0..3 are then specified as follows:

$$y_0 = (29*s_0 + 55*s_1 + s_3 + 64) >> 7 \quad (8-200)$$
$$y_1 = (55*s_2 - 29*s_1 + s_3 + 64) >> 7 \quad (8-201)$$
$$y_2 = (74*s_0 - s_0 - s_2 + 64) >> 7 \quad (8-202)$$
$$y_3 = (55*s_0 + 29*s_2 - s_3 + 64) >> 7 \quad (8-203)$$

[Ed. (WJ): proponent's comment – to share the quantization scaling value, transform precision (7-bit in this case) should be synchronized with DCT.]

Otherwise, the following ordered steps apply: [Ed. (WJ): TBD – 4-point DCT]

8.5.4.2 Transformation process for 8 samples

Inputs of this process are an array of 16 samples of the scaled transform coefficients x with elements x_i, with i = 0..7 and the transform type variable trType.

Output of this process is an array of 16 samples of the residual samples y with elements y_i, with i = 0..7.

[Ed: (WJ) TBD]

8.5.4.3 Transformation process for 16 samples

Inputs of this process are an array of 16 samples of the scaled transform coefficients x with elements x_i, with i = 0..15 and the transform type variable trType.

Output of this process is an array of 16 samples of the residual samples y with elements y_i, with i = 0..15.

[Ed: (WJ) TBD]

8.5.4.4 Transformation process for 32 samples

Inputs of this process are an array of 32 samples of the scaled transform coefficients x with elements x_i, with i=0..31 and the transform type variable trType.

Output of this process is an array of 32 samples of the residual samples y with elements y_i, with i=0..31.

[Ed: (WJ) TBD]

8.5.5 Array construction process

Inputs of this process are:

– a luma location (xB, yB) specifying the top-left luma sample of the current block relative to the top-left luma sample of the current coding unit,

– a variable cIdx specifying the chroma component of the current block,

서식 있음: 들여쓰기: 왼쪽: 0 cm, 내어쓰기: 4.26 글자, 탭: 2.13 글자, 왼쪽
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0.63 cm + 탭 간격: 1.27 cm + 들여쓰기 위치: 1.27 cm, 다음 단락과의 사이에 페이지 나누지 않음, 현재 단락을 나누지 않음, 탭: 1.42 글자, 왼쪽
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 위 첨자/아래 첨자없음
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0.63 cm + 탭 간격: 1.27 cm + 들여쓰기 위치: 1.27 cm, 다음 단락과의 사이에 페이지 나누지 않음, 현재 단락을 나누지 않음, 탭: 1.42 글자, 왼쪽
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 아래 첨자
서식 있음: 들여쓰기: 왼쪽: 0 cm, 내어쓰기: 4.26 글자, 탭: 2.13 글자, 왼쪽

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000402

– a variable nS specifying the size of the current block,

– a variable nCS specifying the size of the current coding unit,

– a (nS)x(nS) array resSampleBlock specifying the current block samples,

– a (nCS)x(nCS) array resSample specifying the samples of the current coding unit.

Output of this process is a modified (nCS)x(nCS) array resSample specifying the samples of the current coding unit.

The modified array resSample is derived as follows.

resSample[xB+i, yB+j] += resSampleBlock[i, j], with i, j = 0..nS-1 (8-~~204~~427)

8.6 In-loop filter process

[Ed: (WJ) overall process description need to be inserted]

8.6.1 Deblocking filter process

A conditional filtering process shall be performed on a treeblock basis after the completion of the picture construction process prior to deblocking filter process for the entire decoded picture (as specified in subclauses XXX and YYY) [Ed.: (WJ) those subclauses seem not defined yet], with all treeblocks in a picture processed in order of increasing treeblock addresses.

Each treeblock is processed on a coding unit basis with the same order as decoding process. For each coding unit, vertical edges are filtered first, starting with the edge on the left-hand side of the coding unit proceeding through the edges towards the right-hand side of the coding unit in their geometrical order. Then, the horizontal edges are filtered starting with the edge on the top of the coding unit proceeding through the edges towards the bottom of the coding unit in their geometrical order.

Sample values above of the current coding unit that may have already been modified by the filtering of horizontal edges of deblocking filter process operation on previous coding unit shall be used as inputs to the deblocking filter process on the current coding unit and may be further modified during the filtering of the current coding unit. Sample values to the left of the current coding unit shall be used as inputs to the deblocking filter process on the current coding unit and may be further modified during the filtering of the current coding unit. Sample values to the left of the current coding unit may be modified by the filtering of vertical edge and may be further modified by the filtering of horizontal edges.

Sample values modified during filtering of vertical edges are used as input for the filtering of the horizontal edges. For sample values modified by both filtering of horizontal edges and filtering of vertical edges, filtering of horizontal edges is applied after filtering of vertical edges.

The deblocking filter process shall be applied to all prediction unit edges and transform unit edges of a picture, except edges at the boundary of the picture ~~and~~, any edges for which the deblocking filter process is disabled by disable_deblocking_filter_idc and any edges coinside with slice boundaries when loop_filter_across_slice_flag is equal to 0. For the transform units and prediction units with edges smaller than 8 samples in either vertical or horizontal direction, only the edges lying on the 8x8 sample grid are filtered.

~~When disable_deblocking_filter_idc is not equal to 1, the deblocking filter process is invoked. If disable_deblocking_filter_idc is equal to 1, the deblocking filter process is simply omitted and if loop_filter_across_slice_flag is equal to 0, the deblocking filter process is omitted at slice boundaries, otherwise as~~ the following ordered steps ~~apply~~ for each coding unit with the same order as decoding process.

1. The coding unit size nS is set equal to 1 << log2CUSize.
2. The variables FilterInternalEdgesFlag, FilterLeftCuEdgeFlag and FilterTopCuEdgeFlag are derived as follows.
   – The variable FilterInternalEdges is set equal to 1.
   – If the left boundary of current coding unit is the left boundary of the picture or if the left boundary of current coding unit is the left boundary of the slice and loop_filter_across_slice_flag is equal to 0, the variable FilterLeftCuEdgeFlag is set equal to 0, otherwise set equal to 1.
   – If the top boundary of current coding unit is the top boundary of the picture or if the top boundary of current coding unit is the top boundary of the slice and loop_filter_across_slice_flag is equal to 0, the variable FilterTopCuEdgeFlag is set equal to 0, otherwise set equal to 1.
3. All elements of two-dimensional array of size (nS)x(nS), horEdgeFlags and verEdgeFlags are initialized to zero.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000403

4. The derivation process of transform unit boundary specified in subclause 8.6.1.1 are invoked with the luma location (xB, yB) set equal to (0, 0), the transform unit size log2TrafoSize set equal to log2CUSize and the variable trafoDepth set equal to 0 as the inputs and the modified horEdgeFlags and verEdgeFlags as outputs.
5. The derivation process of prediction unit boundary specified in subclause 8.6.1.2 are invoked with the coding unit size log2CUSize and the prediction partition mode PartMode as inputs, and the modified horEdgeFlags and verEdgeFlags as outputs.
6. The derivation process of the boundary filtering strength specified in subclause 8.6.1.3 is invoked with the luma location (xC, yC), the coding unit size log2CUSize, horEdgeFlags and verEdgeFlags as inputs and an array of size (2)x(nS)x(nS), bS as output.
7. The filtering process for coding unit specified in subclause 8.6.1.4 are invoked with the luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture, the coding unit size log2CUSize and the array bS as inputs and the modified reconstructued picture as output.

8.6.1.1 Derivation process of transform unit boundary

Inputs of this process are:

– a luma location (xB, yB) specifying the top-left luma sample of the current block relative to the top-left luma sample of the current coding unit,

– a transform unit size log2TrafoSize,

– a variable trafoDepth.

Outputs of this process are:

– two-dimensional arrays of (nS)x(nS), horEdgeFlags and verEdgeFlags.

Depending on split_transform_flag[xB][yB][trafoDepth], the following applies:

– If split_transform_flag[xB][yB][trafoDepth] is equal to 1, the following ordered steps apply:
  1. The variable xB1 is set equal to xB + ((1 << log2TrafoSize) >> 1).
  2. The variable yB1 is set equal to yB + ((1 << log2TrafoSize) >> 1).
  3. The deriviation process of transform unit boundary as specified in this subclause is invoked with the luma location (xB, yB), the variable log2TrafoSize1 set equal to log2TrafoSize – 1 and the variable trafoDepth1 set equal to trafoDepth + 1 as inputs and the outputs are the modified versions of two arrays, horEdgeFlags and verEdgeFlags.
  4. The deriviation process of transform unit boundary as specified in this subclause is invoked with the luma location (xB1, yB), the variable log2TrafoSize1 set equal to log2TrafoSize – 1 and the variable trafoDepth1 set equal to trafoDepth + 1 as inputs and the outputs are the modified versions of two arrays, horEdgeFlags and verEdgeFlags.
  5. The deriviation process of transform unit boundary as specified in this subclause is invoked with the luma location (xB, yB1), the variable log2TrafoSize1 set equal to log2TrafoSize – 1 and the variable trafoDepth1 set equal to trafoDepth + 1 as inputs and the outputs are the modified versions of two arrays, horEdgeFlags and verEdgeFlags.
  6. The deriviation process of transform unit boundary as specified in this subclause is invoked with the luma location (xB1, yB1), the variable log2TrafoSize1 set equal to log2TrafoSize – 1 and the variable trafoDepth1 set equal to trafoDepth + 1 as inputs and the outputs are the modified versions of two arrays, horEdgeFlags and verEdgeFlags.
– Otherwise (split_transform_flag[xB][yB][trafoDepth] is equal to 0), the following applies:
  – If yB is equal to zero, horEdgeFlags[xB + k][yB] is set equal to FilterTopCuEdgeFlag, otherwise horEdgeFlags[xB + k][yB] is set equal to FilterInternalEdgesFlag for k = 0.. (1 << log2TrafoSize) – 1.
  – If xB is equal to zero, verEdgeFlags[xB][yB + k] is set equal to FilterLeftCuEdgeFlag, otherwise verEdgeFlags[xB][yB + k] is set equal to FilterInternalEdgesFlag for k = 0.. (1 << log2TrafoSize) – 1.

8.6.1.2 Derivation process of prediction unit boundary

Inputs of this process are:

– a variable log2CUSize specifying the coding unit size,

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000404

– a prediction partition mode PartMode.

Outputs of this process are:

– two-dimensional arrays of (nS)x(nS), horEdgeFlags and verEdgeFlags.

Depending on PartMode, the following applies:

– If PartMode is equal to PART_2NxN or PART_NxN, horEdgeFlags[k][1 << (log2CUSize – 1)] is set equal to FilterInternalEdgesFlag for k = 0.. (1 << log2CUSize) – 1.

– If PartMode is equal to PART_Nx2N or PART_NxN, verEdgeFlags[1 << (log2CUSize – 1)][k] is set equal to FilterInternalEdgesFlag for k = 0.. (1 << log2CUSize) – 1.

8.6.1.3 Derivation process of boundary filtering strength

Inputs of this process are:

– a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top-left luma sample of the current picture,

– a variable log2CUSize specifying the size of the current coding unit,

– a two-dimensional arrays of size (nS)x(nS), horEdgeFlags and verEdgeFlags.

Output of this process is an array of size (2)x(nS)x(nS), bS specifying the boundary filtering strength.

Let (xE_k, yE_j) with k = 0..nE-1 and j = 0..nE-1 specify a set of edge sample locations where nE is set equal to ((1 << log2CUSize) >> 2), xE_0 = 0, yE_0 = 0, $xE_{k+1} = xE_k + 4$ and $yE_{j+1} = yE_j + 4$.

For (xE_k, yE_j) with k = 0..nE-1 and j = 0..nE-1, the following applies.

– If horEdgeFlags[xE_k][yE_j] is equal to 1,
  – Set sample p_0 = recPicture[xC + xE_k][yC + yE_j – 1] and q_0 = recPicture[xC + xE_k][yC + yE_j].
  – The variable filterDir is set equal to 1.
– Otherwise, if verEdgeFlags[xE_k][yE_j] is equal to 1,
  – Set sample p_0 = recPicture[xC + xE_k – 1][yC + yE_j] and q_0 = recPicture[xC + xE_k][yC + yE_j].
  – The variable filterDir is set equal to 0.
– Depending on the value of filterDir, the variable bS[filterDir][xE_k][yE_j] is derived as follows.
  – If the block edge is also a coding unit edge and the following condition is true, the variable bS[0][xE_k][yE_j] is set equal to 4.
    – The sample p_0 or q_0 is in a coding unit coded with intra prediction mode
  – Otherwise, if the following condition is true, the variable bS[filterDir][xE_k][yE_j] is set equal to 3.
    – The sample p_0 or q_0 is in a coding unit coded with intra prediction mode
  – Otherwise, if the following condition is true, the variable bS[filterDir][xE_k][yE_j] is set equal to 2.
    – The sample p_0 or q_0 is in a transform unit which contains non-zero transform coefficient level.
  – Otherwise, if any of the following conditions are true, the variable bS[filterDir][xE_k][yE_j] is set equal to 1.
    – The prediction unit containing sample p_0 has different reference pictures or a different number of motion vectors with the prediction unit containing the sample q_0.

      NOTE – The determination of whether the reference pictures used for the two prediction are the same or different is based on which pictures are referenced, without regard to whether a prediction is formed using an index into list 0 or an index into list 1, and also without regard to whether or not the index position within a reference picture list is different or not.

    – One motion vector is used to predict the prediction unit containing sample p_0, one motion vector is used to predict the prediction unit containing sample q_0, and the absolute difference between the horizontal or vertical component of the motion vector used is greater than or equal to 4 in units of quarter luma samples.
    – [Ed.: (WJ) needs to be checked again whether this condition covers all 2-motion cases] Two motion vectors are used to predict the prediction unit containing sample p_0, two motion vectors are used to predict

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000405

the prediction unit containing sample q_0, and at least one of the motion vector pairs corresponding the same reference pictures and the different boundary samples p_0 and q_0 satisfies the following condition:

1. The absolute difference between the horizontal or vertical component of a motion vector used in the prediction of the two prediction units is greater than or equal to 4 in units of quarter luma samples.

– Otherwise, the variable bS[filterDir][xE_k][yE_j] is set equal to 0.

8.6.1.4 Filtering process for coding unit

Inputs of this process are:

– a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,

– a variable log2CUSize specifying the coding unit size,

– an array bS specifying the boundary filtering strength.

Output of this process is:

– modified reconstruction of the picture.

The filtering process for luma edges in the current coding unit consists of the following ordered steps:

1. The variable nD is set equal to 1 << (log2CUSize – 3).
2. All elements of the three-dimensional array of size (2)x(nD)x(nD), dEdge are initialized to zero.
3. All elements of the three-dimensional array of size (2)x(nD)x(1<<log2CUSize), dSample are initialized to zero.
4. All elements of the three-dimensional array of size (2)x(nD)x(nD), bStrength ae initialized to zero.
5. For xD_k set equal to xC+(k << 3), k=0..nD – 1, the following applies:
   – For yD_m set equal to yC+(m << 3), m=0..nD – 1, the following ordered steps apply:
     a. Boundary filtering strength bSVer is derived as follows:

        bSVer = Max(bS[0][xD_k][yD_m + i]) for i = 0..7 (8-~~205~~206~~428~~)

     b. bStrength[1][k][m] is set equal to bSVer.
     c. Decision process for luma block edge in subclause 8.6.1.4.1 is invoked with the luma location of the coding unit (xC, yC), the luma location of the block (xD_k, yD_m), a variable verticalEdgeFlag set equal to 1, and the boundary filtering strength bSVer as inputs and the decision dEdge[1][k][m] and an array dS of size (8) as outputs.
     d. dSample[1][k][(m<<3)+i] is set equal to dS[i] for i=0..7.
     e. Boundary filtering strength bSHor is derived as follows:

        bSHor = Max(bS[1][xD_k + i][yD_m]) for i = 0..7 (8-~~206~~207~~428~~)

     f. bStrength[0][k][m] is set equal to bSHor.
     g. Decision process for luma block edge in subclause 8.6.1.4.1 is invoked with the luma location of the coding unit (xC, yC), the luma location of the block (xD_k, yD_m), a variable verticalEdgeFlag set equal to 0, the boundary filtering strength bSHor as inputs, the decision dEdge[0][k][m] and an array dS of size (8) as outputs.
     h. dSample[0][m][(k<<3)+i] is set equal to dS[i] for i=0..7.
6. For xD_k set equal to xC+(k << 3), k=0..nD - 1, the following applies:
   – For yD_m set equal to yB+(m << 3), m=0..nD – 1, the following ordered steps apply:
     a. dS[i] is set equal to dSample[1][k][(m<<3)+i] for i=0..7.
     b. Filtering process for luma block edge in subclause 8.6.1.4.2 is invoked with the luma location of the coding unit (xC, yC), the luma location of the block (xD_k, yD_m), a variable verticalEdgeFlag set equal to 1, the boundary filtering strength bStrength[1][k][m], the decision dEdge[1][k][m], and the array of size (8), dS as inputs and the modified luma picture buffer as outputs.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000406

7. For yD_m set equal to yC+(m << 3), m=0..nD - 1, the following applies:
   - For xD_k set equal to xC+(k << 3), k=0..nD – 1, the following ordered steps apply:
     a. If xD_k is equal to 0, the parameter xPOS is set equal to 1. If xD_k is equal to xB+((nD – 1) << 3) xPOS is set equal to 2. Otherwise xPOS is set to 0.
     b. dS[i] is set equal to dSample[0][m][(k << 3) + i] for i = 0..7.
     c. Filtering process for luma block edge in subclause 8.6.1.4.2 is invoked with the luma location of the coding unit (xC, yC), the luma location of the block (xD_k, yD_m), a variable verticalEdgeFlag set equal to 0, the boundary filtering strength bStrength[0][k][m], the decision dEdge[0][k][m], and the array of size (8), dS, xPOS, dS_L[m][], dE_L[m], bS_L[m], and t_{CL}[m], as inputs and the modified luma picture buffer as output.
   - The elements of the two dimensional array of size (3)x(nD), dS_L are set as follows. dS_L[m][0], dS_L[m][1], and dS_L[m][2], are set equal to dS[5], dS[6] and dS[7].
   - The elements of the array of size (nD), dE_L are set as follows. dE_L[m] is set equal to dEdge[0][k][m].
   - The elements of the array of size (nD), bS_L are set as follows. bS_L[m] is set equal to bStrength[0][k][m].
   - The elements of the array of size (nD), t_C are set as follows. t_{CL}[m] is set equal to t_c.

The filtering process for chroma edges in the current coding unit consists of the following ordered steps:

1. The variable nD is set equal to 1 << (Max(log2CUSize, 4) – 4).
2. For xD_k set equal to (xC / 2)+(k << 3), k=0..nD – 1, the following applies:
   - For yD_m set equal to (yC / 2)+(m << 2), m=0..nD*2 – 1, the following ordered steps apply:
     a. Boundary filtering strength bSVer is derived as follows:

$$bSVer = bS[\ 0\ ][\ xD_k*2\ ][\ yD_m*2\ ] \quad (8\text{-}207439)$$

     b. Filtering process for chroma block edge in subclause 8.6.1.4.3 is invoked with the chroma location of the coding unit (xC/2, yC/2), the chroma location of the block (xD_k, yD_m), a variable verticalEdgeFlag set equal to 1, a chroma component index cIdx set equal to 1 and the boundary filtering strength bSVer as inputs and the modified chroma picture buffer as output.
     c. Filtering process for chroma block edge in subclause 8.6.1.4.3 is invoked with the chroma location of the coding unit (xC/2, yC/2), the chroma location of the block (xD_k, yD_m), a variable verticalEdgeFlag set equal to 1, a chroma component index cIdx set equal to 2 and the boundary filtering strength bSVer as inputs and the modified chroma picture buffer as output.
3. For yD_m set equal to (yC / 2)+(m << 3), m=0..nD – 1, the following applies:
   - For xD_k set equal to (xC / 2)+(k << 2), k=0..nD*2 – 1, the following ordered steps apply:
     a. If xD_k is equal to 0, the parameter xPOS is set equal to 1. If xD_k is equal to xB+((nD*2 – 1) << 2) xPOS is set equal to 2. Otherwise xPOS is set to 0.
     b. Boundary filtering strength bSHor is derived as follows:

$$bSHor = bS[\ 1\ ][\ xD_k*2\ ][\ yD_m*2\ ] \quad (8\text{-}208434)$$

     c. Filtering process for chroma block edge in subclause 8.6.1.4.3 is invoked with the chroma location of the coding unit (xC/2, yC/2), the chroma location of the block (xD_k, yD_m), a variable verticalEdgeFlag set equal to 0, a chroma component index cIdx set equal to 1 and the boundary filtering strength bSHor, xPOS, bS_L[m] and t_{CL}[m] as inputs and the modified chroma picture buffer as output.
     d. Filtering process for chroma block edge in subclause 8.6.1.4.3 is invoked with the chroma location of the coding unit (xC/2, yC/2), the chroma location of the block (xD_k, yD_m), a variable verticalEdgeFlag set equal to 0, a chroma component index cIdx set equal to 2 and the boundary filtering strength bSHor, xPOS, bS_L[m] and t_{CL}[m] as inputs and the modified chroma picture buffer as output.

서식 있음: 제목 5

8.6.1.4.1 Decision process for luma block edge

Inputs of this process are:

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000407

– a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,

– a luma location (xB, yB) specifying the top-left luma sample of the current block relative to the top left luma sample of the current coding unit,

– a variable verticalEdgeFlag,

– a variable bS specifying the boundary filtering strength,

Output of this process is:

– a variable dE containing a decision,

– one-dimensional array of size (8), dS containing decisions.

Let s' represent the luma sample array $recPicture_L$ of the current picture.

A variables β is specified as Table 8-14~~Table 8-13~~ with luma quantization parameter qP_L as input.

A variable t_C is specified as follows:

– If bS is greater than 2, the variable t_C is specified specified as Table 8-14~~Table 8-13~~ with luma quantization parameter (qP_L + 4) as input,

– Otherwise (bS is equal or less than 2), the variable t_C is specified specified as Table 8-14~~Table 8-13~~ with luma quantization parameter qP_L as input.

Depending on verticalEdgeFlag, the following applies:

– If verticalEdgeFlag is equal to 1, the following ordered steps apply:

1. The sample values $p_{i,k}$ and $q_{i,k}$ with i = 0..3 and k = 2,5 are derived as follows:

$$q_{i,k} = s'[\ xC + xB + i, yC + yB + k\ ] \quad (8\text{-}209\text{~~447~~})$$

$$p_{i,k} = s'[\ xC + xB - i - 1, yC + yB + k\ ] \quad (8\text{-}210\text{~~448~~})$$

2. The variable d is derived as follows:

$$d = |\ p_{2,2} - 2*p_{1,2} + p_{0,2}\ | + |\ q_{2,2} - 2*q_{1,2} + q_{0,2}\ | + |\ p_{2,5} - 2*p_{1,5} + p_{0,5}\ | + |\ q_{2,5} - 2*q_{1,5} + q_{0,5}\ | \quad (8\text{-}211\text{~~449~~})$$

3. The variable dE is set equal to 0.

4. If bS is not equal to 0 and d is less than β, the following ordered steps apply:

   a. for each sample location (xC + xB, yC + yB + k), k = 0..7, the following ordered steps apply:

      a. The decision process for a luma sample specified in subclause 8.6.1.4.4 is invoked with sample values $p_{i,k}$, $q_{i,k}$ with i = 0..3, the boundary filtering strength bS and the variables d, β and t_C as inputs and a decision dSam as output.

      b. The variable dS[k] is set equal to dSam

   b. The variable dE is set equal to 1.

– Otherwise (verticalEdgeFlag is equal to 0), the following ordered steps apply:

1. The sample values $p_{i,k}$ and $q_{i,k}$ with i = 0..3 and k = 2,5 are derived as follows:

$$q_{i,k} = s'[\ xC + xB + k, yC + yB + i\ ] \quad (8\text{-}212\text{~~435~~})$$

$$p_{i,k} = s'[\ xC + xB + k, yC + yB - i - 1\ ] \quad (8\text{-}213\text{~~436~~})$$

2. The variable d is derived as follows:

$$d = |\ p_{2,2} - 2*p_{1,2} + p_{0,2}\ | + |\ q_{2,2} - 2*q_{1,2} + q_{0,2}\ | + |\ p_{2,5} - 2*p_{1,5} + p_{0,5}\ | + |\ q_{2,5} - 2*q_{1,5} + q_{0,5}\ | \quad (8\text{-}214\text{~~437~~})$$

3. The variable dE is set equal to 0.

4. If bS is not equal to 0 and d is less than β, the following ordered steps apply:

   a. For each sample location (xC + xB + k, yC + yB), k = 0..7, the following ordered steps apply:

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000408

a. The decision process for a luma sample specified in subclause 8.6.1.4.4 is invoked with sample values $p_{i,k}$, $q_{i,k}$ with i = 0..3, the boundary filtering strength bS and the variables d, β and t_C as inputs and a decision dSam as output.

b. The variable dS[k] is set equal to dSam.

b. The variable dE is set equal to 1.

8.6.1.4.3 8.6.1.4.2 **Filtering process for luma block edge**

Inputs of this process are:

– a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,

– a luma location (xB, yB) specifying the top-left luma sample of the current block relative to the top left luma sample of the current coding unit,

– a variable verticalEdgeFlag,

– a variable bS specifying the boundary filtering strength,

– a variable dE containing a decision,

– one-dimensional array of size (8), dS containing decisions,

– a variable bS_L,

– a variable t_{CL},

Output of this process is:

– modified reconstruction of the picture.

Let s' represent the luma sample array $recPicture_L$ of the current picture.

Depending on pcm_flag, a variable β is specified as follows:

– If pcm_flag is equal to 1, A the variables β is specified as Table 8-14Table 8-11 with luma quantization parameter Q as input.

– Otherwise, the variables β is specified as Table 8-14Table 8-11 with luma quantization parameter qP_L as input.

서식 있음: 들여쓰기: 왼쪽: 0 cm, 내어쓰기: 2.84 글자, 탭: 1.42 글자, 왼쪽

A variable t_C is specified as follows:

– If bS is greater than 2, the variable t_C is specified as Table 8-14Table 8-11 with luma quantization parameter (qP_L + 4) as input,

– Otherwise (bS is equal or less than 2), the variable t_C is specified as Table 8-14Table 8-11 with luma quantization parameter qP_L as input.

Depending on verticalEdgeFlag, the following applies:

– If verticalEdgeFlag is equal to 1, the following ordered steps apply:

1. The sample values $p_{i,k}$ and $q_{i,k}$ with i = 0..3 and k = 0..7 are derived as follows:

$$q_{i,k} = s'[\ xC + xB + i, yC + yB + k\ ] \quad (8\text{-}215\text{-}432)$$

$$p_{i,k} = s'[\ xC + xB - i - 1, yC + yB + k\ ] \quad (8\text{-}216\text{-}433)$$

2. If dE is not equal to 0, for each sample location (xC + xB, yC + yB + k), k = 0..7, the following ordered steps apply:

   a. The filtering process for a luma sample specified in subclause 8.6.1.4.5 is invoked with sample values $p_{i,k}$, $q_{i,k}$ with i = 0..3, the decision dS[k], the boundary filtering strength bS and the variable t_C as inputs and the number of filtered samples nD and the filtered sample values p_i' and q_i' as outputs.

   b. The filtered sample values p_i' and q_i' with i = 0..nD – 1 replace the corresponding samples inside the sample array s' as follows:

$$s'[\ xC + xB + i, yC + yB + k\ ] = q_i' \quad (8\text{-}217\text{-}435)$$

$$s'[\ xC + xB - i - 1, yC + yB + k\ ] = p_i' \quad (8\text{-}218\text{-}436)$$

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000409

– Otherwise (verticalEdgeFlag is equal to 0), the following ordered steps apply:

1. If xPOS is equal to 1, the parameters k_s and k_e are set to -3 and 4 respectively. If xD is equal to 2, the parameters k_s and k_e are set to 0 and 4 respectively. Otherwise k_s and k_e are set to 0 and 7 respectively.
2. The sample values $p_{i,k}$ and $q_{i,k}$ with i = 0..3 and k = k_s..k_e are derived as follows:

$$q_{i,k} = s'[\ xC + xB + k, yC + yB + i\ ] \quad (8\text{-}219437)$$

$$p_{i,k} = s'[\ xC + xB + k, yC + yB - i - 1\ ] \quad (8\text{-}220438)$$

3. If xPOS is less than 0 and dE_L is not equal to 0, for each sample location (xC + xB + k, yC + yB), k = -3..-1, the following ordered steps apply:
   a. The filtering process for a luma sample specified in subclause 8.6.1.4.5 is invoked with sample values $p_{i,k}$, $q_{i,k}$ with i = 0..3, decision $dS_L[k+3]$, the boundary filtering strength bS_L and the variable t_{CL} as inputs and the number of filtered samples nD and the filtered sample values p_i' and q_i' as outputs.
   b. The filtered sample values p_i' and q_i' with i = 0..nD – 1 replace the corresponding samples inside the sample array s' as follows:

$$s'[\ xC + xB + k, yC + yB + i\ ] = q_i' \quad (8\text{-}221439)$$

$$s'[\ xC + xB + k, yC + yB - i - 1\ ] = p_i' \quad (8\text{-}222469)$$

5. If dE is not equal to 0, for each sample location (xC + xB + k, yC + yB), k = 0.. k_e , the following ordered steps apply:
   a. The filtering process for a luma sample specified in subclause 8.6.1.4.5 is invoked with sample values $p_{i,k}$, $q_{i,k}$ with i = 0..3, decision dS[k], the boundary filtering strength bS and the variable t_C as inputs and the number of filtered samples nD and the filtered sample values p_i' and q_i' as outputs.
   b. The filtered sample values p_i' and q_i' with i = 0..nD – 1 replace the corresponding samples inside the sample array s' as follows:

$$s'[\ xC + xB + k, yC + yB + i\ ] = q_i' \quad (8\text{-}223464)$$

$$s'[\ xC + xB + k, yC + yB - i - 1\ ] = p_i' \quad (8\text{-}224463)$$

8.6.1.4.3 Filtering process for chroma block edge

서식 있음: 제목 5

[Ed.: (WJ) cIdx cannot be 0 here]

Inputs of this process are:

– a luma location (xC, yC) specifying the top-left chroma sample of the current coding unit relative to the top left chroma sample of the current picture,
– a luma location (xB, yB) specifying the top-left chroma sample of the current block relative to the top left chroma sample of the current coding unit,
– a variable verticalEdgeFlag,
– a variable bS specifying the boundary filtering strength,
– a variable cIdx specifying the chroma component index.
– a variable xPOS,
– a variable bS_L,
– a variable t_{CL}

Output of this process is:

– modified reconstruction of the picture.

Let s' be a variable specifying chroma sample array which is derived as follows.

– If cIdx is equal to 1, s' represents the chroma sample array $recPicture_{Cb}$ of the current picture.
– Otherwise (cIdx is equal to 2), s' represents the chroma sample array $recPicture_{Cr}$ of the current picture.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000410

A variable t_C is specified as follows:

– If bS is greater than 2, the variable t_C is specified specified as Table 8-14~~Table 8-13~~ with luma quantization parameter ($qP_L + 4$) as input,

– Otherwise (bS is equal or less than 2), the variable t_C is specified as Table 8-14~~Table 8-13~~ with luma quantization parameter qP_L as input.

Depending on verticalEdgeFlag, the following applies:

– If verticalEdgeFlag is equal to 1, for each sample location (xC + xB, yC + yB + k), k = 0..3, the following ordered steps apply:

1. The sample values p_i and q_i with i = 0..1 are derived as follows:

$$q_i = s'[\ xC + xB + i, yC + yB + k\ ] \quad (8\text{-}225\sout{443})$$

$$p_i = s'[\ xC + xB - i - 1, yC + yB + k\ ] \quad (8\text{-}226\sout{443})$$

2. If bS is greater than 2, the following ordered steps apply:

   a. The filtering process for a sample specified in subclause 8.6.1.4.6 is invoked with sample values p_i, q_i, with i = 0..1, the boundary filtering strength bS and the variable t_C as inputs and the filtered sample values p_0' and q_0' as outputs.

   b. The filtered sample values p_0' and q_0' replace the corresponding samples inside the sample array s' as follows:

$$s'[\ xC + xB , yC + yB + k\ ] = q_0' \quad (8\text{-}227\sout{444})$$

$$s'[\ xC + xB - 1, yC + yB + k\ ] = p_0' \quad (8\text{-}228\sout{445})$$

– Otherwise (verticalEdgeFlag is equal to 0), the following ordered steps apply:

1. If xPOS is equal to 1, the parameters k_s and k_e are set to -1 and 2 respectively. If xD is equal to 2, the parameters k_s and k_e are set to 0 and 2 respectively. Otherwise k_s and k_e are set to 0 and 3 respectively.

2. The sample values p_i and q_i with i = 0..1 and k = k_s..k_e are derived as follows:

$$q_i = s'[\ xC + xB + k, yC + yB + i\ ] \quad (8\text{-}229\sout{446})$$

$$p_i = s'[\ xC + xB + k, yC + yB - i - 1\ ] \quad (8\text{-}230\sout{447})$$

3. If xPOS is less than 0, and if bS_L, is greater than 2, for each sample location (xC + xB - 1, yC + yB), the following ordered steps apply:

   a. The filtering process for a sample specified in subclause 8.6.1.4.6 is invoked with sample values p_i, q_i, with i = 0..1, the boundary filtering strength bS_L and the variable t_{CL} as inputs and the filtered sample values p_0' and q_0' as outputs.

   b. The filtered sample values p_0' and q_0' replace the corresponding samples inside the sample array s' as follows:

$$s'[\ xC + xB + k, yC + yB\ ] = q_0' \quad (8\text{-}231\sout{452})$$

$$s'[\ xC + xB + k, yC + yB - i - 1\ ] = p_i' \quad (8\text{-}232\sout{445})$$

4. If bS is greater than 2, for each sample location (xC + xB + k, yC + yB), k = 0.. k_e, the following ordered steps apply:

   c. The filtering process for a sample specified in subclause 8.6.1.4.6 is invoked with sample values p_i, q_i, with i = 0..1, the boundary filtering strength bS and the variable t_C as inputs and the filtered sample values p_0' and q_0' as outputs.

   d. The filtered sample values p_0' and q_0' replace the corresponding samples inside the sample array s' as follows:

$$s'[\ xC + xB + k, yC + yB\ ] = q_0' \quad (8\text{-}233\sout{448})$$

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000411

$$s'[\ xC + xB + k, yC + yB - 1\ ] = p_0' \quad (8\text{-}234449)$$

8.8.3.4.38.6.1.4.5 Decision process for a luma sample

[Ed: (WJ) no filtering when bS is equal to 0]

Inputs of this process are:

- sample values, p_i and q_i with i = 0..2,
- a variable bS specifying the boundary filtering strength,
- variables d, β and t_C.

Output of this process is:

- a variable dSam containing a decision

When the variable bS is not equal to 0, the following applies:

- If d is less than (β >> 2), | $p_3 - p_0$ | + | $q_0 - q_3$ | is less than (β >> 3) and | $p_0 - q_0$ | is less than (5*t_C + 1) >> 1, dSam is set equal to 1.
- Otherwise, dSam is set equal to 0.

8.8.3.4.38.6.1.4.6 Filtering process for a luma sample

Inputs of this process are:

- sample values, p_i and q_i with i = 0..3,
- a variable dSam containing a decision,
- a variable t_C.

Output of this process is:

- number of filtered samples nD,
- filtered sample values, p_i' and q_i' with i = 0..nD – 1,

Depending on dSam, the following applies:

- When the variable dSam is equal to 1, the following strong filtering applies while nD is set equal to 3:

$$p_0' = Clip1_Y((p_2 + 2*p_1 + 2*p_0 + 2*q_0 + q_1 + 4) >> 3) \quad (8\text{-}235450)$$

$$p_1' = Clip1_Y((p_2 + p_1 + p_0 + q_0 + 2) >> 2) \quad (8\text{-}236451)$$

$$p_2' = Clip1_Y((2*p_3 + 3*p_2 + p_1 + p_0 + q_0 + 4) >> 3) \quad (8\text{-}237452)$$

$$q_0' = Clip1_Y((p_1 + 2*p_0 + 2*q_0 + 2*q_1 + q_2 + 4) >> 3) \quad (8\text{-}238453)$$

$$q_1' = Clip1_Y((p_0 + q_0 + q_1 + q_2 + 2) >> 2) \quad (8\text{-}239454)$$

$$q_2' = Clip1_Y((p_0 + q_0 + q_1 + 3*q_2 + 2*q_3 + 4) >> 3) \quad (8\text{-}240455)$$

- Otherwise, the following weak filtering applies while nD is set equal to 2:

$$\Delta = Clip3(-t_C, t_C, (13*(q0 - p0) + 4*(q1 - p1) - 5*(q2 - p0) + 16) >> 5) \quad (8\text{-}241456)$$

$$p_0' = Clip1_Y(p_0 + \Delta) \quad (8\text{-}242457)$$

$$q_0' = Clip1_Y(q_0 - \Delta) \quad (8\text{-}243458)$$

$$p_1' = Clip1_Y(p_1 + \Delta/2) \quad (8\text{-}244459)$$

$$q_1' = Clip1_Y(q_1 - \Delta/2) \quad (8\text{-}245460)$$

Each of the filtered sample values, p_i' with i = 0..nD-1, is substituted by the corresponding input sample value p_i if all of the following conditions are true.

서식 있음: 아래 첨자

서식 있음: 아래 첨자

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000412

— p_i is a sample of an I_PCM block.

— pcm_loop_filter_disable_flag value is equal to 1.

Similary, each of the filtered sample values, q_i' with i = 0..nD-1, is substituted by the corresponding input sample value q_i if all of the following conditions are true.

— q_i is a sample of an I_PCM block.

— pcm_loop_filter_disable_flag value is equal to 1.

[Ed. (WJ): for PCM case, deblocking filter applies first and the filtered pixels are restored. Rather than this, it's better to skip the filtering itself for PCM samples since first filtering is actually not needed.]

Table 8-~~148~~ ~~128~~ ~~13~~ – Derivation of threshold variables β and t_C from input Q

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7	8
t_C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Q	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
β	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28	30	32	34	36
t_C	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4
Q	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	
β	38	40	42	44	46	48	50	52	54	56	58	60	62	64	64	64	64	64	
t_C	5	5	6	6	7	8	9	9	10	10	11	11	12	12	13	13	14	14	

~~8.6.1.4.8~~ ~~8.6.1.4.6~~ 8.6.1.4.6 Filtering process for a chroma sample

[Ed: (WJ) no filtering when bS is equal or less than 2]

Inputs of this process are:

– sample values, p_i and q_i with i = 0..1,

– a variable bS specifying the boundary filtering strength.

– a variable t_C.

Output of this process is:

– The filtered sample values, p_0' and q_0'.

When the variable bS is greater than 2, the filtered sample values p_0' and q_0' are derived by

$$\Delta = \text{Clip3}(-t_C, t_C, ((((q_0 - p_0) << 2) + p_1 - q_1 + 4) >> 3)) \quad (8\text{-}~~246~~464)$$

$$p_0' = \text{Clip1}_C(p_0 + \Delta) \quad (8\text{-}~~247~~462)$$

$$q_0' = \text{Clip1}_C(q_0 - \Delta) \quad (8\text{-}~~248~~463)$$

The filtered sample value, p_0' is substituted by the corresponding input sample value p_0 if all of the following conditions are true.

— p_0 is a sample of an I_PCM block.

— pcm_loop_filter_disable_flag value is equal to 1.

Similary, the filtered sample value, q_0' is substituted by the corresponding input sample value q_0 if all of the following conditions are true.

— q_0 is a sample of an I_PCM block.

— pcm_loop_filter_disable_flag value is equal to 1.

서식 있음: 들여쓰기: 왼쪽: 0.5 cm, 내어쓰기: 2.84 글자, 탭: 2.84 글자, 왼쪽

서식 있음: 아래 첨자

서식 있음: 아래 첨자

서식 있음: 표준, 들여쓰기: 내어쓰기: 2.84 글자, 탭: 2.84 글자, 왼쪽 + 4.26 글자(없음) + 5.67 글자 + 7.09 글자

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000413

[Ed. (WJ): for PCM case, deblocking filter applies first and the filtered pixels are restored. Rather than this, it's better to skip the filtering itself for PCM samples since first filtering is actually not needed.]

서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자

8.6.2 Sample adaptive offset process

A sample adaptive offset process shall be conditionally performed after the completion of the deblocking filter process for the decoded picture.

This process is invoked when both sample_adaptive_offset_enabled_flag and sample_adaptive_offset_flag are equal to 1.

This process is performed on a region basis which is aligned with the LCU boundaries after the completion of the slice construction process prior to sample adaptive offset process for the entire decoded process.

The sample adaptive offset process is invoked with the region equal to entire slice here.

[Ed.: (WJ) invoking is needed]

8.6.2.1 Modification process for sample adaptive offset

[Ed.: (WJ) recursive partitioning is invoked here]

8.6.2.1.1 Modification process for luma samples

[Ed:. (WJ) offset is added here]

8.6.3 Adaptive loop filter process

An adaptive loop filtering process shall be conditionally performed on a treeblock basis after the completion of the sample adaptive offset process for the entire decoded picture, with all treeblocks in a picture processed in order of increasing treeblock addresses.

Each treeblock is processed on a coding unit basis with the same order as decoding process.

This process is invoked when both adaptive_loop_filter_enabled_flag and adaptive_loop_filter_flag are equal to 1.

This process is performed on a coding unit basis after the completion of the slice construction process prior to adaptive loop filter process for the entire decoded slice, with all coding units in a slice processed in order of coding unit scan order.

Filter coefficients c_L for luma samples and c_C for chroma samples are derived by invoking the process specified in subclause 8.6.3.2~~8.6.2.1~~.

Filter index array fIdx for luma samples are derived by invoking the process specified in subclause 8.6.3.3~~8.6.2.2~~ with the luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture and the coding unit size log2CUSize as inputs.

When interpreting the luma samples of the coding unit as to be filtered, depending on pcm_flag, pcm_loop_filter_disable_flag and alf_cu_control_flag, the following applies.

- If pcm_loop_filter_disable_flag is equal to 1 and pcm_flag is equal to 1, the luma samples are not filtered.
- Otherwise, ~~If~~ if alf_cu_control_flag is equal to 0, the luma samples are filtered,
- Otherwise (alf_cu_control_flag is equal to 1), if AlfCuFlag[xC][yC] is equal to 1 where xC and yC are the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture, the luma samples are filtered.
- Otherwise, the luma samples are not filtered.

When interpreting the chroma samples of the coding unit as to be filtered, depending on pcm_flag, pcm_loop_filter_disable_flag and alf_chroma_idc, the following applies.

- If pcm_loop_filter_disable_flag is equal to 1 and pcm_flag is equal to 1, both chroma samples are not filtered.
- Otherwise, ~~If~~ if alf_chroma_idc is equal to 1, the Cr samples are filtered.
- Otherwise, if alf_chroma_idc is equal to 2, the Cb samples are filtered.
- Otherwise, if alf_chroma_idc is equal to 3, both chroma samples are filtered.
- Otherwise (alf_chroma_idc is equal to 0), both chroma samples are not filtered.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000414

For the luma samples of the coding unit interpreted as to be filtered, the filtering process for luma samples specified in subclause 8.6.3.4~~8.6.2.3~~ is invoked with the luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture, the coding unit size log2CUSize, and the filter index array fIdx as inputs and the output is the modified filtered picture, $recFiltPicture_L$.

For the chroma samples of the coding unit interpreted as to be filtered, the filtering process for chroma samples specified in subclause 8.6.3.5~~8.6.2.4~~ is invoked with the chroma location (xC/2, yC/2) specifying the top-left chroma sample of the current coding unit relative to the top left chroma sample of the current picture, the coding unit size log2CUSize-1 and the chroma component index cIdx equal to 1 as inputs and the output is the modified filtered picture, $recFiltPicture_{Cb}$.

For the chroma samples of the coding unit interpreted as to be filtered, the filtering process for chroma samples specified in subclause 8.6.3.5~~8.6.2.4~~ is invoked with the chroma location (xC/2, yC/2) specifying the top-left chroma sample of the current coding unit relative to the top left chroma sample of the current picture, the coding unit size log2CUSize-1 and the chroma component index cIdx equal to 2 as inputs and the output is the modified filtered picture, $recFiltPicture_{Cr}$.

[Ed.: (WJ) recPicture: deblocked/output picture and recFiltPicture: ALFed picture]

[Ed.: (WJ) depending adaptive_loop_filter_flag, recFiltPicture should be copied to recPicture in subclause 8.6]

8.6.3.1 Boundary padding process

Inputs of this process are:

- a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,
- a variable log2CUSize specifying the size of the current coding unit,
- a variable cIdx specifying the chroma component index,

Output of this process is the padded luma sample array *s''*.

Depending on cIdx, the following applies:

- If cIdx is equal to 0, a luma sample array *s'* is set equal to $recPicture_L$, a variable alfTap is set equal to (alf_length_luma_minus_5_div2 << 1) + 5, a variable *nS* is set equal to (1 << log2CUSize) and a variable nExtPixels is set equal to (alfTap >> 1).
- Otherwise, if cIdx is equal to 1, a chroma sample array *s'* is set equal to $recPicture_{Cb}$, a variable alfTap is set equal to (alf_length_chroma_minus_5_div2 << 1) + 5, a variable *nS* is set equal to (1 << (log2CUSize – 1)) and a variable nExtPixels is set equal to (alfTap >> 1).
- Otherwise (cIdx is equal to 2), a chroma sample array *s'* is set equal to $recPicture_{Cr}$, a variable alfTap is set equal to (alf_length_chroma_minus_5_div2 << 1) + 5, a variable *nS* is set equal to (1 << (log2CUSize – 1)) and a variable nExtPixels is set equal to (alfTap >> 1).

Depending on loop_filter_across_slice_flag, a variable *s''* specifying the padded luma sample array is derived as:

- If loop_filter_across_slice_flag is equal to 1, *s''*[xC+x][yC+y] is set equal to *s'*[xC+x][yC+y] for x = -nExtPixels..nS+nExtPixels-1 and y = -nExtPixels..nS+nExtPixels-1.

  [Ed. (WJ): picture boundaries should be considered even for this case]

- Otherwise (loop_filter_across_slice_flag is equal to 0), *s''*[xC+x][yC+y] is derived by the following ordered steps:

  1. *s''*[xC+x][yC+y] is set equal to *s'*[xC+x][yC+y] for x=0..nS – 1 and y=0..nS – 1.
  2. If the left boundary of current block coinsides the slice,

     s''[xC+x][yC+y] = *s'*[xC][yC+y] where x = -nExtPixels..-1 and y = 0..nS-1 (8-249462)

  3. If the right boundary of current block coinsides the slice,

     s''[xC+nS+x][yC+y] = *s'*[xC+nS-1][yC+y] where x = 0..nExtPixels-1 and y = 0..nS-1 (8-250462)

  4. If the top boundary of current block coinsides the slice,

     s''[xC+x][yC+y] = *s'*[xC+x][yC] where x = 0..nS-1 and y = -nExtPixels..-1 (8-251462)

  5. If the bottom boundary of current block coinside the slice,

서식 있음: 제목 4,Heading 4 Char1,Heading 4 Char Char

서식 있음: 글머리 기호 + 수준:1 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm, 탭: 3.97 글자(없음)

서식 있음: 아래 첨자

서식 있음: 아래 첨자

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000415

s''[xC+ x][yC+nS+y] = s'[xC+x][yC+nS-1] where x = 0..nS-1 and y = 0..nExtPixels-1 (8-252463)

6. If both the left and the top boundaries of current block coinside the slice and the current coding unit and its upper-left coding unit are not at the same slice,

s''[xC+x][yC+y] = s'[xC][yC] where x = -nExtPixels..-1 and y = -nExtPixels..-1 (8-253462)

7. If both the right and the top boundaries of current block coinside the slice and the current coding unit and its upper-right coding unit are not at the same slice,

s''[xC+nS+x][yC+y] = s'[xC+nS-1][yC] where x = 0..nExtPixels-1 and y = -nExtPixels..-1 (8-254462)

8. If both the left and the bottom boundaries of current block coinside the slice and the current coding unit and its bottom-left coding unit are not at the same slice,

s''[xC+x][yC+nS+y] = s'[xC][yC+nS-1] where x = -nExtPixels..-1 and y = 0..nExtPixels-1 (8-255462)

9. If both the right and the bottom boundaries of current block coinside the slice and the current codinx unit and its bottom-right coding unit are not at the same slice,

s''[xC+nS+x][yC+nS+y] = s'[xC+nS-1][yC+nS-1] where x = 0..nExtPixels-1 and y = 0..nExtPixels-1 (8-256462)

8.6.3.3.8.3.2 **Derivation process for filter coefficients**

Outputs of this process are filter coefficients c_L for the luma samples and filter coefficients c_C for the chroma samples.

The luma filter coefficients c_L with elements $c_L[i][j]$, i = 0..AlfNumFilters – 1, j = 0..AlfCodedLengthLuma – 1 is derived as follows:

– If alf_pred_method is equal to 0 or the value of i is equal to 0,

$c_L[i][j] = alf\_coeff\_luma[i][j]$ (8-257464)

– Otherwise (alf_pred_method is equal to 1 and the value of i is greater than 1),

$c_L[i][j] = alf\_coeff\_luma[i][j] + c_L[i – 1][j]$ (8-258465)

Considering the symmetry of the filter, the luma filter coefficients c_L with elements $c_L[i][j]$, i = 0..AlfNumFilters – 1 is derived as follows:

$c_L[i][AlfLengthLuma – 1] = c_L[i][AlfCodedLengthLuma – 1]$ (8-259466)

$c_L[i][AlfLengthLuma – 2 – j] = c_L[i][j]$ for j = 0..AlfCodedLengthLuma – 2 (8-260467)

The chroma filter coefficients c_C with elements $c_C[i]$, i = 0..AlfCodedLengthChroma – 1 is derived as follows:

– If i is equal to AlfCodedLengthChroma – 1, the coefficient $c_C[i]$ is derived as

$c_C[i] = 255 – sum – alf\_coeff\_chroma[i]$ (8-261468)

where

$sum = alf\_coeff\_chroma[AlfCodedLengthChroma – 2] + \Sigma_j(alf\_coeff\_chroma[j] << 1)$ (8-262469)

with j = 0..AlfCodedLengthChroma – 3

– Otherwise (i is less than AlfCodedLengthChroma – 1),

$c_C[i] = alf\_coeff\_chroma[i]$ (8-263470)

Considering the symmetry of the filter, the chroma filter coefficients c_C with elements $c_C[i]$, i = 0..AlfLengthChroma – 1 is derived as follows:

서식 있음: 들여쓰기: 왼쪽: 0.54 cm, 번호 매기기 + 수준:1 + 번호 스타일: 1, 2, 3, … + 시작 번호: 1 + 맞춤: 왼쪽 + 맞춤 위치: 0 cm + 탭 간격: 0.71 cm + 들여쓰기 위치: 0.71 cm, 탭: 3.6 글자, 왼쪽 + 5.4 글자, 왼쪽 + 7.2 글자, 왼쪽 + 8.51 글자, 왼쪽 + 1.42 글자(없음) + 3.97 글자 + 5.96 글자 + 7.94 글자 + 9.93 글자

서식 있음: (한글) 한국어, (언어2) (없음)

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000416

$c_C[\ i\ ][\ AlfLengthChroma - 1\ ] = c_C[\ i\ ][\ AlfCodedLengthChroma - 1\ ]$ (8-264474)

$c_C[\ i\ ][\ AlfLengthChroma - 2 - j\ ] = c_C[\ i\ ][\ j\ ]$ for $j = 0..AlfCodedLengthChroma - 2$ (8-265473)

서식 있음: 제목 4,Heading 4 Char1,Heading 4 Char Char, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자

~~8.6.3.28.6.3.3~~ **Derivation process for filter index array for luma samples**

Inputs of this process are:

- a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,
- a variable log2CUSize specifying the size of the current coding unit.

Output of this process is the two-dimensional filter index array of (nS)x(nS), fIdx.

A variable nS is set equal to (1 << log2CUSize) ~~and s'' represents the luma sample array reconPicture$_L$ of the current picture~~.

The boundary padding process specified in subclause 8.6.3.1 is invoked with the luma location (xC, yC), the size of coding unit log2CUSize and the chroma component index cIdx set equal to 0, and the output is assigned to the luma sample array s''. {Ed. (WJ): s'' is now a picture-size array, but actually CU size + appropriate margin is enough}

The filter index array fIdx is specified in the follows:

- When alf_region_adaptation_flag is equal to 1, the following ordered steps apply.

  1. The variables xIdx and yIdx are derived as

$regionTab[16] = \{ 0, 1, 4, 5, 15, 2, 3, 6, 14, 11, 10, 7, 13, 12, 9, 8 \}$ (8-266474)
$offset = 1 << (Log2MaxCUSize - 1)$ (8-267474)
$xInterval = (((PicWidthInSamples_L + offset) >> Log2MaxCUSize) + 1) >> 2$ (8-268474)
$yInterval = (((PicHeightInSamples_L + offset) >> Log2MaxCUSize) + 1) >> 2$ (8-269474)
$xIdx = Min(3, Floor((xC + x) / (xInterval << Log2MaxCUSize)))$ (8-270474)
$yIdx = Min(3, Floor((yC + y) / (yInterval << Log2MaxCUSize)))$ (8-271474)

  2. The filter index fIdx[x, y] with x, y = 0..(nS)-1 is derived as

$fIdx[x][y] = regionTab[(yIdx << 2) + xIdx]$ (8-272474)

- Otherwise (alf_region_flag is equal to 0), the following ordered steps apply.

  3. The variables varTempH[x][y], varTempV[x][y] and varTempl[x][y] with x, y = -1..(nS)+1 is derived as

$varTempH[x][y] = | (s''[xC+x, yC+y] << 1) - s''[xC+x-1, yC+y] - s''[xC+x+1, yC+y] |$ (8-273474)
$varTempV[x][y] = | (s''[xC+x, yC+y] << 1) - s''[xC+x, yC+y-1] - s''[xC+x, yC+y+1] |$ (8-274474)
$varTempl[x][y] = varTempH[x][y] + varTempV[x][y]$ (8-275474)

  4. The variable varTemp2[x, y] with x, y = 0..(nS)-1 is derived as

$varTemp2[x][y] = \Sigma_i\Sigma_j\ varTempl[x + i][y + j]$ with i, j = -1..1 (8-276474)

  5. The variables varTemp3[x, y], varTempH1[x, y] and varTempV1[x, y] with x, y = 0..((nS) – 1)>>2 are derived as

$varTemp3[x][y] = (\Sigma_i\Sigma_j\ varTemp2[(x << 2) + i][(y << 2) + j]) >> 4$ with i, j = 0..3 (8-277474)
$varTempH1[x][y] = \Sigma_i\Sigma_j\ varTempH[(x << 2) + i][(y << 2) + j]$ with i, j = 0..3 (8-278474)
$varTempV1[x][y] = \Sigma_i\Sigma_j\ varTempV[(x << 2) + i][(y << 2) + j]$ with i, j = 0..3 (8-279474)

  6. The variable direction is derived as

     – If varTempV1[x >> 2][y >> 2] is greater than varTempH1[x >> 2][y >> 2] << 1,

       direction = 1

     – Otherwise, if varTempH1[x >> 2][y >> 2] is greater than varTempV1[x >> 2][y >> 2] << 1,

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000417

direction = 2

– Otherwise,

direction = 0

7. The variable avgvar is derived as

varTab[16] = { 0, 1, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4 } (8-280475)
avgVar = Clip3(0, 15, (varTemp3[x >> 2][y >> 2] * 114) >> (3 + BitDepthY)) (8-281475)

8. The filter index fIdx[x, y] with x, y = 0..(nS)-1 is derived as

fIdx[x][y] = Clip3(0, 4, varTab[avgVar]) + 5 * direction (8-282304)

8.6.3.38.6.3.4 **Filtering process for luma samples**

Inputs of this process are:

– a luma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,

– a variable log2CUSize specifying the size of the current coding unit,

– a filter index array of (nS)x(nS), fIdx.

Output of this process is the filtered reconstruction of luma picture.

~~Let s'' be a variable specifying luma sample array recPicture.~~The boundary padding process specified in subclause 8.6.3.1 is invoked with the luma location (xC, yC), the size of coding unit log2CUSize and the chroma component index cIdx set equal to 0, and the output is assigned to the luma sample array s''. [Ed. (WJ): s'' is now a picture-size array, but actually CU size + appropriate margin is enough]

A variable nS is set equal to (1 << log2CUSize) and a variable alfTap is set equal to (alf_length_luma_minus_5_div2 << 1) + 5.

Each sample of luma picture recFiltPicture$_L$[xC + x][yC + y] with x, y = 0..(nS)-1, is derived as follows.

$$recFiltPicture_L[xC+x][yC+y] = c_L[fIdx[x][y]][N] + \sum_{i=0}^{N-1}\left(s''[xC+x+horPos[i], yC+y+verPos[i]] * c_L[fIdx[x][y]][i]\right) \quad (8\text{-}283476)$$

where N is set equal to AlfNumCoeffLuma-1 and horPos[i] and verPos[i] are specified in Table 8-15~~Table 8-14~~ and Table 8-16~~Table 8-15~~, respectively.

Table 8-158.13.13 – Specification of horPos[i] according to alfTap for adaptive loop filter process of luma samples

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
alfTap = 5	0	-1	0	1	-2	-1	0	1	2	-1	0	1	0	-	-	-	-	-	-	-
alfTap = 7	0	-1	0	1	-2	-1	0	1	2	-3	-2	-1	0	1	2	3	-2	-1	0	1
alfTap = 9	-1	0	1	-2	-1	0	1	2	-3	-2	-1	0	1	2	3	-4	-3	-2	-1	0

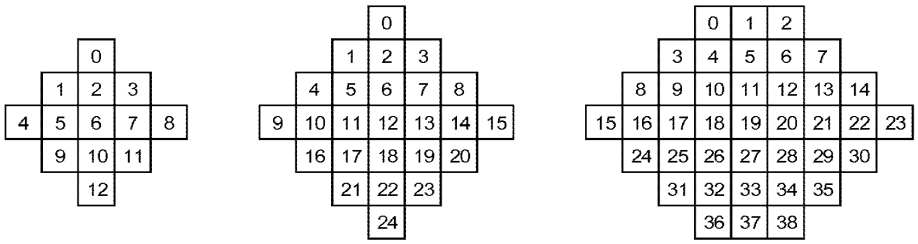
Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000418

i	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
alfTap = 5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
alfTap = 7	2	-1	0	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-
alfTap = 9	1	2	3	4	-3	-2	-1	0	1	2	3	-2	-1	0	1	2	-1	0	1

Table 8-168-14 – Specification of verPos[i] according to alfTap for adaptive loop filter process of luma samples

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
alfTap = 5	-2	-1	-1	-1	0	0	0	0	0	1	1	1	2							
alfTap = 7	-3	-2	-2	-2	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	1	1	1	1
alfTap = 9	-3	-3	-3	-2	-2	-2	-2	-2	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0

i	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
alfTap = 5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
alfTap = 7	1	2	2	2	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
alfTap = 9	0	0	0	0	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3



(a) alfTap = 5 (a) alfTap = 7 (a) alfTap = 9

Figure 8-8 Mapping between geometric position and luma adaptive loop filter index according to alfTap (informative)

8.8.3.48.8.3.8 **Filtering process for chroma samples**

Inputs of this process are:

- a chroma location (xC, yC) specifying the top-left luma sample of the current coding unit relative to the top left chroma sample of the current picture,
- a variable log2CUSize specifying the size of the current coding unit.
- a variable cIdx specifying the chroma component index.

Output of this process is the filtered reconstruction of chroma picture.

The boundary padding process specified in subclause 8.6.3.1 is invoked with the chroma location (xC, yC), the size of coding unit log2CUSize and the chroma component index cIdx, and the output is assigned to the luma sample array s'', [Ed. (WD): s'' is now a picture-size array, but actually CU size + appropriate margin is enough]

~~Let s' be a variable specifying chroma sample array which is derived as follows.~~

~~If cIdx is equal to 1, s' represents the chroma sample array recPicture$_{Cb}$ of the current picture and recFiltPicture represents the filtered reconstruction of chroma picture recFiltPicture$_{Cb}$.~~

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000419

~~Otherwise (cIdx is equal to 2), s' represents the chroma sample array $recPicture_{C_r}$ of the current picture and recFiltPicture represents the filtered reconstruction of chroma picture $recFiltPicture_{C_r}$.~~

A variable nS is set equal to (1 << log2CUSize) and a variable alfTapChroma is set equal to (alf_length_chroma_minus_5_div2 << 1) + 5.

Filtered samples of chroma picture recFiltPicture[xC + x][yC + y] with x, y = 0..(nS)-1, are derived as follows.

$$recFiltPicture[xC + x][yC + y] = c_C[N] + \sum_{i=0}^{N-1} (s''[\, xC + x + horPos[i], yC + y + verPos[i]\,] * c_C[i]) \quad (8\text{-}284477)$$

where

$$N = AlfNumCoeffChroma - 1, \quad (8\text{-}285478)$$

$$horPos[\, i\,] = (\, i \,\%\, alfTapChroma\,) - (\, alfTapChroma >> 1\,), \text{ and} \quad (8\text{-}286479)$$

$$verPos[\, i\,] = (\, i \,/\, alfTapChroma\,) - (\, alfTapChroma >> 1\,) \quad (8\text{-}287480)$$

서식 있음: (한글) 한국어, (언어2) (없음)

서식 있음: 표준, 들여쓰기: 왼쪽: 0 cm, 탭: 4.26 글자(없음) + 5.67 글자 + 7.09 글자

9 Parsing process

Inputs to this process are bits from the RBSP.

Outputs of this process are syntax element values.

This process is invoked when the descriptor of a syntax element in the syntax tables in subclause 7.3 is equal to ue(v), me(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), me(v), se(v), or te(v). For syntax elements in subclauses 7.3.4 and 7.3.5, this process is invoked only when entropy_coding_mode_flag is equal to 0.

Inputs to this process are bits from the RBSP.

Outputs of this process are syntax element values.

Syntax elements coded as ue(v), me(v), or se(v) are Exp-Golomb-coded. Syntax elements coded as te(v) are truncated 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 0. This process is specified as follows:

```
leadingZeroBits = −1
for( b = 0; !b; leadingZeroBits++ )                    (9-1)
    b = read_bits( 1 )
```

The variable codeNum is then assigned as follows:

$$codeNum = 2^{leadingZeroBits} - 1 + read\_bits(\, leadingZeroBits\,) \quad (9\text{-}2)$$

where the value returned from read_bits(leadingZeroBits) is interpreted as a binary representation of an unsigned integer with most significant bit written first.

Table 9-1~~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 that 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~~Table 9-1~~. The "suffix" bits are those bits that are parsed in the computation of codeNum and are shown as x_i in Table 9-1~~Table 9-1~~, with i being in the range 0 to leadingZeroBits − 1, inclusive. Each x_i can take on values 0 or 1.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000420

Table 9-18-3 – Bit strings with "prefix" and "suffix" bits and assignment to codeNum ranges (informative)

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 9-2Table 9-2 illustrates explicitly the assignment of bit strings to codeNum values.

Table 9-28-3 – Exp-Golomb bit strings and codeNum in explicit form and used as ue(v) (informative)

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
...	...

Depending on the descriptor, the value of a syntax element is derived as follows.

- If the syntax element is coded as ue(v), the value of the syntax element is equal to codeNum.
- Otherwise, if the syntax element is coded as se(v), the value of the syntax element is derived by invoking the mapping process for signed Exp-Golomb codes as specified in subclause 9.1.1 with codeNum as the input.
- Otherwise, if the syntax element is coded as me(v) [Ed. (TW): insert text]
- Otherwise (the syntax element is coded as te(v)), the range of possible values for the syntax element is determined first. The range of this syntax element may be between 0 and x, with x being greater than or equal to 1 and the range is used in the derivation of the value of the syntax element value as follows.
  - If x is greater than 1, codeNum and the value of the syntax element is derived in the same way as for syntax elements coded as ue(v).
  - Otherwise (x is equal to 1), the parsing process for codeNum which is equal to the value of the syntax element is given by a process equivalent to:

```
b = read_bits( 1 )                                    (9-3)
codeNum = !b
```


Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000421

9.1.1 Mapping process for signed Exp-Golomb codes

Input to this process is codeNum as specified in subclause 9.1.

Output of this process is a value of a syntax element coded as se(v).

The syntax element is assigned to the codeNum 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~~Table 9-3~~ provides the assignment rule.

Table 9-3~~3~~ – Assignment of syntax element to codeNum for signed Exp-Golomb coded syntax elements se(v)

codeNum	syntax element value
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 slice data

This process is invoked when the descriptor of a syntax element in the syntax tables in subclause 7.3 is equal to ce(v).

Inputs to this process are bits from the RBSP.

Outputs of this process are syntax element values.

9.2.1 Parsing process for VLC codes

Inputs to this process are bits from the RBSP and the parameter vlcNum specifying the VLC code. If vlcNum is equal to 14, inputs to this process additionally include a parameter cMax.

Output of this process is the variable codeNum specifying an integer value to be used when deriving syntax elements.

If vlcNum is not equal to 8 or 11, and is less than 14, the parsing process for the VLC codes 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 0. This process is specified as follows.

```
leadingZeroBits = −1
for( b = 0; !b; leadingZeroBits++ )                    (9-4)
    b = read_bits( 1 )
```

Depending on the value of vlcNum, the value of codeNum is assigned as follows.

– If vlcNum is equal to 0,

  – If leadingZeroBits is less than or equal to 6,

```
codeNum = leadingZeroBits                              (9-5)
```

  – Otherwise (leadingZeroBits is greater than 6),

```
numBits = leadingZeroBits − 6                          (9-6)
codeNum = 5 + (1<< numBits) + read_bits( numBits )
```

– Otherwise, if vlcNum is less than 5,

  – If leadingZeroBits is less than or equal to 6,

```
codeNum = (leadingZeroBits << vlcNum) + read_bits( vlcNum )   (9-7)
```


Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000422

- Otherwise (leadingZeroBits is greater than 6),

```
numBits = leadingZeroBits – 6 + vlcNum                                   (9-8)
codeNum = 5*(1<<vlcNum) + (1<< numBits) + read_bits( numBits)
```

- Otherwise, if vlcNum is less than 8,

```
codeNum = (leadingZeroBits << (vlcNum - 4)) + read_bits( vlcNum - 4)     (9-9)
```

- Otherwise, if vlcNum is equal to 8,

```
b = read_bits(1)                                                         (9-10)
codeNum = b ? 2 - read_bits(1) : 0
```

- Otherwise, if vlcNum is equal to 9,
  - If leadingZeroBits is equal to 0,

```
b = read_bits(1)
if (b) {
    b = read_bits(1)
    if (b) {
        codeNum = 3 + read_bits(3)                                       (9-11)
    }
    else{
        b = read_bits(1)
        codeNum = b ? 1 + read_bits(1) : 0
    }
}
```

  - Otherwise (leadingZeroBits is greater than 0),

```
codeNum = (leadingZeroBits << 4) + read_bits( 4) + 11                    (9-12)
```

- Otherwise, if vlcNum is equal to 10,

```
codeNum = (1<< leadingZeroBits) − 1 + read_bits( leadingZeroBits )       (9-13)
```

- Otherwise, if vlcNum is equal to 11,

```
codeNum = read_bits(3)
if( codeNum ) {
    b = read_bits(1)                                                     (9-14)
    codeNum = (codeNum << 1) + b -1
}
```

- Otherwise, if vlcNum is equal to 12,

```
codeNum = (leadingZeroBits << 6) + read_bits( 6 )                        (9-15)
```

- Otherwise, if vlcNum is equal to 13,

```
codeNum = (leadingZeroBits << 4) + read_bits( 4 )                        (9-16)
```

- Otherwise, if vlcNum is equal to 14,

```
leadingZeroBits = −1
for( b = 0; !b && leadingZeroBits < cMax; leadingZeroBits++ )            (9-17)
    b = read_bits( 1 )
codeNum = leadingZeroBits
```

- Otherwise, if vlcNum is equal to 15, codeNum is determined according to Table 9-4.
- Otherwise, if vlcNum is equal to 16

```
codeNum = read_bits(3)
if( codeNum >= 6)
    codeNum = 7 - codeNum
else if (codeNum > 0) {                                                  (9-18)
    b = read_bits( 1 )
    codeNum = 13 – (codeNum << 1) - b
}
else {
    b = read_bits( 2 )
```


Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000423

```
        codeNum = 15 – b
    }
```

– Otherwise, if vlcNum is equal to 17

```
    codeNum = read_bits(3)
    if( codeNum ) {                                                    (9-19)
        b = read_bits(1)
        codeNum = 16 -  (codeNum << 1) - b
    }
```

– Otherwise, if vlcNum is equal to 18

```
    codeNum = read_bits(3)
    if( codeNum && codeNum < 4) {
        b = read_bits(1)
        codeNum = 8 -  (codeNum << 1) - b
    } else if (codeNum < 6) {
        b = read_bits(2)
        codeNum = 30 -  (codeNum << 2) - b                             (9-20)
    } else {
        b = read_bits(3)
        codeNum =  (codeNum << 3) + b
        if (codeNum < 62)
            codeNum = 76 – codeNum
        else {
            b = read_bits(1)
            codeNum = 156 - (codeNum << 1) – b
        }
    }
```

– Otherwise, if vlcNum is equal to 19

```
    codeNum = read_bits(3)
    if( codeNum && codeNum < 3) {
        b = read_bits(1)
        codeNum = 6 -  (codeNum << 1) - b
    } else {                                                           (9-21)
        b = read_bits(2)
        codeNum =  (codeNum << 2) + b
        if (codeNum < 25)
            codeNum = 29 – codeNum
        else {
            b = read_bits(1)
            codeNum = 81 - (codeNum << 1) – b
        }
    }
```

Table 9-[illegible] – codeNum and codeword with vlcNum equal to 15

codeNum	Codeword
0	0
1	11
2	1001
3	1011
4	1000
5	10100
6	101010
7	101011

9.2.2 Initialisation process

Outputs of this process are initialised CAVLC internal variables.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000424

The processes in subclauses 9.2.2.2 through 9.2.2.6 are invoked when starting the parsing of the slice data of a slice in subclause 7.3.4.

9.2.2.1 Counter initialisation for codeword index adaptive mapping

Outputs of this process are initial values of counters that are used for parsing a syntax element.

If the number of counters used for a syntax element is N (N>0), values of the N counters are all initialized to 0. For each set of counters used for parsing a syntax element, there is an associated value, counterSum, to store the number of counter increments since its previous reset. The value of counterSum is also initialized to 0.

9.2.2.2 Initialisation process for lastPosVlcNumIndex

Outputs of this process are initial values of the variable array lastPosVlcNumIndex.

The variable array lastPosVlcIndex is initialized as follows.

lastPosVlcNumIndex [blockType] = 0, for blockType = 0...7 (9-2224)

9.2.2.3 Initialisation process for lastPosTable

Outputs of this process are initial values of the variable array lastPosTable.

The variable array lastPosTable is initialized as specified in Table 9-5.

Table 9-58 – Specification of initial values of lastPosTable[tableNum][codeNum]

	codeNum															
tableNum	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	0	1	2	3	5	4	7	6	11	8	14	9	16	15	10	13
1	0	1	2	5	7	4	6	3	9	12	11	8	10	14	13	15
2	0	1	2	5	7	4	6	3	9	12	11	8	10	14	13	15

	codeNum															
tableNum	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
0	12	17	18	19	22	31	23	21	27	20	25	30	24	26	28	29
1	16	17	21	22	25	18	28	23	30	26	20	19	27	24	29	31
2	16	17	21	22	25	18	28	23	30	26	20	19	27	24	29	31

9.2.2.4 Initialisation process for splitPredPartModeTable

Outputs of this process are initial values of the variable array splitPredPartModeTable and associated counters.

The variable array splitPredPartModeTable is initialized as follows:

– For each cuDepth with a value from 0 to (Log2MaxCUSize – Log2MinCUSize), set splitPredPartModeTable[cuDepth][codeNum] equal to codeNum, with codeNum = 0…6.

For each cuDepth with a value from 0 to (Log2MaxCUSize – Log2MinCUSize), there are four counters used for codeword index adaptive mapping. These counters are initialized according to subclause 9.2.2.1.

9.2.2.5 Initialisation process for intraModeTable

Outputs of this process are initial values of the variable array intraModeTable.

The variable array intraModeTable is initialized as specified in Table 9-6.

Amazon.com, Inc., Amazon.com Services LLC - Ex. 1002, Page 000425