Sub-Sequence Video Coding For Improved Temporal Scalability

Dong Tian
Tampere International Center for Signal Processing
Tampere, Finland
dong.tian@tut.fi

Miska M. Hannuksela
Nokia Research Center
Tampere, Finland
miska.hannuksela@nokia.com

Moncef Gabbouj
Tampere University of Technology
Tampere, Finland
moncef.gabbouj@tut.fi

***Abstract*—Compression efficiency and bitrate scalability are among the key factors in video coding. The paper introduces novel sub-sequence coding techniques for temporal scalability. The presented coding schemes provide a wider range for bitrate scaling than conventional temporal scalability methods and maintain high coding efficiency at the same time. The proposed sub-sequence techniques are adopted into the latest video coding standard H.264, making it easy to identify sub-sequences and possible to discard them intentionally. As shown by the extensive simulations, a wide range of applications, from mobile messaging to consumer electronics such as digital TV can benefit from sub-sequences.**

I. Introduction

In recent years, scalable video coding has been one of the key challenges in the field of video coding. Scalable bitstreams can be used for various purposes, such as adjustment of the transmitted bitrate according to the prevailing network throughput in streaming applications and scaling the complexity of the decoding process according to the available computational resources. Scalable coding also partitions the coded bitstream into sections with different impact on decoded video quality. These sections can be used in the transport layer to implement unequal error protection. Scalable video coding methods can be classified into temporal, spatial, and SNR techniques, as well as any combination of them.

Two general categories exist for interframe coding in temporal scalable video coding algorithms: predictive coding and subband coding [1]. All prevailing video coding standards, such as H.263, H.264 (aka MPEG-4 AVC), MPEG-2 Visual, and MPEG-4 Visual, deploy motion compensation predictive techniques, and hence this paper focuses on the temporal scalability for predictive coding.

The paper introduces a novel sub-sequence coding technique, which is an enhancement of the known temporal scalability methods. It is shown that the range for bitrate scaling is wider and the compression efficiency is the same or better compared to earlier methods. Thus, the proposed method gives more flexibility in applications utilizing bitrate scalability, such as rate scaling in streaming servers.

Modern video coding techniques often utilize multiple reference pictures for motion compensation to improve compression efficiency and error resilience. The sub-sequence technique also makes use of multiple reference pictures. A typical mode for reference pictures operation is "sliding window", which removes the oldest reference frame from the buffer when a new reference frame is decoded and the buffer is full.

This paper is organized as follows. Section II reviews the conventional temporal scalable coding. The proposed sub-sequence technique and coding schemes for improved temporal scalability are given in Section III. Section IV discusses the simulation results. Finally, we conclude the work in Section V.

II. Conventional Temporal Scalability

A. Individually Disposable Pictures

In other video coding standards than H.264, bi-predictive (B) pictures are not used as prediction references. Consequently, they provide a way to achieve temporal scalability.

The enhanced reference picture selection mode (Annex U) of H.263 allows signaling whether a particular picture is a reference picture for any inter prediction of any other picture. Consequently, a picture not used for prediction (a non-reference picture) can be safely disposed. The H.264 syntax

0-7803-8834-8/05/$20.00 ©2005 IEEE. 6074
Authorized licensed use limited to: Penn State University. Downloaded on April 01,2025 at 17:45:30 UTC from IEEE Xplore. Restrictions apply.

APPLE 1046

includes similar signaling to distinguish between reference and non-reference pictures.

B. Disposal of Picture Chains

A known method in today's streaming systems to cope with drastically dropped channel throughput is to transmit Intra pictures only. When the network throughput is restored, Inter pictures can be transmitted again from the beginning of the next Group of Pictures (GOP).

Generally, any chain of Inter pictures can be safely disposed, if no other picture is predicted from them. This fact can be utilized to treat Inter pictures at the end of a prediction chain as less important than other Inter pictures. The known layered coding techniques put some pictures into enhancement layers for temporal scalability, but do not identify the dependencies of pictures. In addition, multiple prediction chains are often maintained to achieve temporal scalability. In the conventional solutions, it is hard for the server or gateway to discard pictures intentionally without affecting the decoder behavior.

III. Sub-Sequences And H.264

A. Sub-Sequence and Sub-Sequence Layer

The proposed sub-sequence represents a number of inter-dependent pictures that can be disposed without affecting the decoding of any other sub-sequence in the same sub-sequence layer or any sub-sequence in any lower sub-sequence layer. The sub-sequence technique enables easy identification of disposable chains of pictures when processing pre-coded bitstreams.

Disposal of a sub-sequence on which there are no dependencies in the bitstream maintains a valid bitstream. Thus, the decoding process for the remaining bitstream and the reference picture buffer handling in particular has to be such that it does not depend on the presence or absence of any disposable sub-sequences. Subsection III.C describes the fundamentals how the decoding process of H.264 takes sub-sequences into consideration.

Pictures in a coded bitstream can be organized into sub-sequences and sub-sequence layers in multiple ways provided that the structure fulfills the requirements for dependencies between sub-sequences and sub-sequence layers. In most applications, a single structure of sub-sequences and sub-sequence layers is sufficient. Each picture belongs to exactly one sub-sequence, and each sub-sequence belongs to exactly one sub-sequence layer in any sub-sequence structure.

Sub-sequence layers are arranged hierarchically based on their dependency on each other. The base layer (layer 0) is independently decodable. Sub-sequence layer 1 depends on some of the data in layer 0, i.e., correct decoding of all pictures in sub-sequence layer 1 requires decoding of all the previous (in decoding order) pictures in layer 0. In general, correct decoding of sub-sequence layer N requires decoding of layers from 0 to N-1. It is recommended to organize sub-sequences into sub-sequence layers in such a way that discarding of enhanced layers results in a constant or nearly constant picture rate. Picture rate and therefore subjective quality increase along with the number of decoded sub-sequence layers.

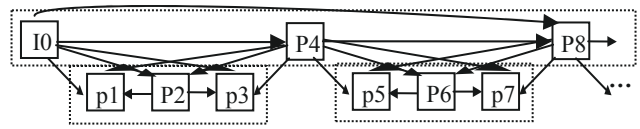

Figure 1. Example of sub-sequences: coding pattern "IpPpP" (*The numbers in the figure indicates the output order and the number of reference frames is 3.*)

Compared with the conventional layered scalability, sub-sequences can be a non-layered (i.e. one-layer) bitstream with no added complexity on handling multiple layers. Sub-sequence technique enables easy identification of independent sub-sequences within the layers, making the bitrate shaping more efficient.

Since a sub-sequence in the base layer can be decoded independently of any other sub-sequences, the beginning of a base layer sub-sequence can be used as a random access position.

B. Use of Sub-Sequences

Sub-sequences can be used for improved bitrate scalability and error resiliency. Improved bitrate scalability can be achieved without sacrificing compression efficiency. In this sub-section, we present the sub-sequence coding scheme for improved bitrate scalability. We also discuss how the fast forward operation can be improved with the proposed sub-sequence scheme. The use of sub-sequences in error resilience has been demonstrated at least in [4] and [5] and we do not discuss the topic here.

Fig. 1 illustrates an example of a sub-sequence coding scheme referred to as IpPpP within H.264 codec. 'P' and 'p' denote reference picture and non-reference picture, respectively. The decoding order of pictures is as follows: I0 P4 P2 p1 p3 P8 P6 p5 p7. The midmost P picture in IpPpP is not used as a reference picture for pictures other than the two p pictures in the same sub-sequence. Any non-reference picture (p picture) can be safely discarded. Any sub-sequence pPp can be discarded without affecting the decoding of other sub-sequences pPp. A modification of the sub-sequence coding scheme IpPpP is to replace the P and p in sub-sequence layer 1 to B and b, respectively. Noting that B pictures can also be used as references in H.264 (see subsection III.C).

There are at least two methods that are often used with the conventional GOP structure (referred to as IbbP in this paper) for the fast forward operation: decoding only the I pictures of each GOP and decoding only the I and P pictures. The proposed sub-sequence scheme (IbBbP) provides an additional method for the fast forward operation: decoding

Authorized licensed use limited to: Penn State University. Downloaded on April 01,2025 at 17:45:30 UTC from IEEE Xplore. Restrictions apply.

only the reference pictures in layer 0. In other words, the IbBbP scheme enables one additional fast forward speed in player implementations.

C. Sub-Sequences in H.264

1) Overview

H.264 includes three main differences in the concept of P and B pictures and their relation to reference picture buffering when compared to previous video standards such as H.263. First, both a P slice and a B slice allow using multiple reference pictures to predict sample values. However, each block in P slices can only use at most one motion vector, whereas each block in B slices can use at most two motion vectors. Second, whether a picture is a reference picture is indicated independently from the slices types, which implies that a B picture can be stored as a reference picture as well. Third, the decoding order of pictures is totally decoupled with their output (presenting) order. Thus, the decoded picture buffer is not only for buffering reference pictures but also for storing such non-reference pictures that are output with a delay.

2) Gaps in frame number

Frame number (the frame_num syntax element in the slice header) is used to identify different reference frames. By monitoring the continuity of frame numbers, decoders can detect losses of reference frames. Further actions can be invoked upon the founding of gaps in frame numbers. However, when a streaming server or a gateway disposes a sub-sequence intentionally, an H.264 decoder should not infer any frames losses. Instead, the decoder inserts "non-existing" frames into the decoded picture buffer as if the frames with absent frame numbers were decoded normally. Only when any "non-existing" frames are referred in the following decoding process, unexpected frame losses can be deduced.

3) Sub-sequences related SEI messages

Supplemental enhancement information (SEI) is data embedded in the coded bitstream that is not required for correct decoding of the sample values. However, SEI messages may help the decoder at least in displaying the decoded pictures or concealing transmission errors. Three types of SEI messages are defined for sub-sequences. The sub-sequence information SEI message maps a coded picture to a certain sub-sequence and sub-sequence layer. The sub-sequence layer characteristics SEI message and the sub-sequence characteristics SEI message give statistical information, such as bitrate, on the indicated sub-sequence layer and sub-sequence respectively. Furthermore, the dependencies between sub-sequences are indicated in the sub-sequence characteristics SEI message. Decoders can use these messages to scale the decoding process computationally in case of lack of computational resources and to detect in which sub-sequences and sub-sequence layers accidentally lost pictures (during transmission) resided, and thus improve error resilience.

4) File format

Information on sub-sequences and sub-sequence layering can be included in the file format specified for H.264 [3]. The file format is based on the ISO base media file format and can be used as an extension of the MP4 file format, for example. As consequences, streaming servers can easily adapt the bitrate of the transmitted streams by deciding which sub-sequence layers and sub-sequences are transmitted. File players can use the sub-sequence information for the implementation of the fast forward operation.

IV. Simulations

A. Simulation Environment

To evaluate the coding performance of IpPpP and IbBbP, they were compared with IPPP, IppP and IbbP within H.264 codec. In IPPP, all the Inter pictures are P pictures. In IppP, the two p pictures are non-reference pictures predicted from both the previous frames and the subsequent frame in output order. In IbbP, the two b pictures are non-reference pictures.

To demonstrate the usefulness of the proposed technique to a variety of applications, such as mobile messenger and digital TV, we carried out simulations for the following picture sizes and frame rates: QCIF 15 Hz, QCIF 30 Hz, CIF 30 Hz, and 525SD 25 Hz. The size of the decoded picture buffer was selected according to level 1 (QCIF), level 2 (CIF) and level 3 (525SD) of H.264. As the decoded picture buffer stores also the non-reference frames whose output is delayed, the number of reference frames (the size of the "sliding window" for reference pictures) for IpPpP and IbBbP is one less than that for IPPP, IppP and IbbP. The number of reference frames in each case is listed in Table I.

We used a constant quantization parameter (QP) value for all pictures in sub-sequence layer 0. In sub-sequence layer 1, we used a constant QP value that is 2 units larger than the QP value in the base layer. We coded each original sequence six times, QP values for layer 0 pictures being 20, 24, 28, 32, 36 and 40.

B. Marking Reference Pictures

The midmost P picture in IpPpP was not used as a reference picture after the decoding of the second p picture. Memory management control operation (MMCO) command in H.264 allows marking a reference picture to be unused for reference. Since MMCO commands can only be associated to reference pictures, we assigned a MMCO command to P8 to mark P2 to be unused for reference (when the notation as of Fig. 1 is used). P6 was marked to be unused for reference at P12, and so on. Similar MMCO commands were used in IbBbP too.

C. Simulation Results

We ran simulations to compare the rate-distortion performance of different coding schemes at full frame rate. The rate-distortion curve of Paris in CIF at 30Hz is shown in Fig. 2 as an example. Bjontegaard delta PSNR [6] was used

Authorized licensed use limited to: Penn State University. Downloaded on April 01,2025 at 17:45:30 UTC from IEEE Xplore. Restrictions apply.

to evaluate the average differences between rate-distortion curves. Table II contains the Bjontegaard delta PSNR values of the three competitive pairs: IpPpP vs. IPPP, IpPpP vs. IppP and IbBbP vs. IbbP. A positive value implies the former scheme outperforms the latter. It can be found that the compression performance of IpPpP is very close to that of IppP and IbBbP even outperforms IbbP a little in most cases.

The comparisons of H.264 Main/Extended profile with the Baseline profile, i.e., IbbP vs. IppP or IbBbP vs. IpPpP, are also presented in Table II. We can easily see the superiority of B and b pictures over P and p pictures regarding the compression efficiency.

The share of bits allocated for sub-sequence layer 0 and all reference pictures is shown in Table III. It can be seen that the proposed sub-sequence schemes provide a larger range to adapt the bitrate of a transmitted or decoded bitstream. Moreover, the proposed sub-sequence schemes provide two steps of bitrate scalability that result into a constant picture rate, whereas the IbbP and IppP schemes provide only one such step. On the average, the IpPpP coding scheme provides bitrate steps at constant frame rate at about 50% and about 70% of the full bitrate, whereas the IppP coding scheme can be scaled down to an average of 60% of the full bitrate while maintaining constant frame rate. Similarly, the IbBbP coding scheme provides bitrate steps of about 60% and 80% of the full bitrate, whereas decoding of the reference frames in the IbbP coding scheme results into an average of 70% of the full bitrate.

V. CONCLUSIONS

This paper proposes a novel sub-sequence coding technique which can be applied to any video coding standards with multiple reference pictures buffer. IpPpP and IbBbP are proposed to provide more scalability compared to IPPP, IppP, and IbbP patterns while maintaining at least as high coding efficiency. We presented how sub-sequences are adopted in H.264, including the decoding process on gaps of frame number, sub-sequence related SEI messages and file format for H.264. Finally, the extensive simulations show the improvement in performance compared to conventional schemes.

REFERENCES

[1] G. J. Conklin, S. S. Hemami, "A Comparison of Temporal Scalability Techniques," IEEE Trans on CSVT, vol. 9, no. 6, pp. 909-919, Sept 1999.

[2] T. Wiegand, G. Sullivan and A. Luthra, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification," document JVT-G050r1, May 2003

[3] D. Singer and T. Walker, "Study Text of Amd7 of ISO/IEC 14496-1 PDAM7," MPEG-4 Systems, N5096, Aug 2002

[4] S. Wenger, "Video Redundancy Coding in H.263+," PV 1997

[5] M. M. Hannuksela, "Simple Packet Loss Recovery Method for Video Streaming," PV2001, South Korea, May 2001

[6] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T VCEG-M33, March 2001

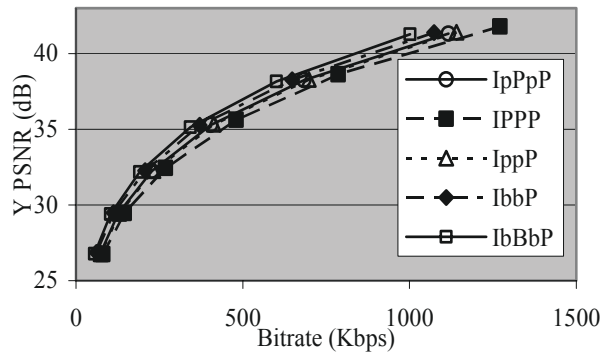

Figure 2. Rate-distortion curves for Pairs (*CIF @ 30 Hz*)

TABLE I. NUMBER OF REFERENCE FRAMES

-	IpPpP, IbBbP	IPPP, IppP, IbbP
QCIF	3	4
CIF	5	6
SD	4	5

TABLE II. AVERAGE RATE-DISTORTION DIFFERENCES (DB) AT FULL FRAME RATE (a: IpPpP vs. IPPP, b: IpPpP vs. IppP, c: IbBbP vs. IbbP, d: IppP vs. IbbP, e: IpPpP vs. IbBbP. A positive value implies the former scheme outperforms the latter)

Sequences		a	b	c	d	e
QCIF 15Hz	***Foreman***	0.07	-0.11	-0.01	-0.27	-0.37
	Paris	0.55	-0.08	0.08	-0.46	-0.63
	Tempete	0.32	-0.09	0.10	-0.45	-0.63
QCIF 30Hz	***Foreman***	0.27	-0.15	-0.00	-0.43	-0.58
	Paris	0.69	-0.06	0.29	-0.65	-1.02
	Container	1.03	0.16	0.25	-0.79	-0.90
CIF 30Hz	***Mobile***	0.62	-0.00	0.21	-0.74	-0.95
	Paris	0.63	0.09	0.84	-0.54	-0.74
	Tempete	0.47	0.05	0.21	-0.45	-0.61
SD 25Hz	***Mobile***	0.05	-0.12	0.20	-0.85	-1.18
	Parkrunner	0.03	-0.10	0.23	-0.59	-0.92

TABLE III. BITRATE PERCENTAGES AT LOWER FRAME RATES (%). (The fraction in the column titles (1/2, 1/3, 1/4) indicates the picture rate compared to the full picture rate.)

Sequences		IpPpP		IppP	IbBbP		IbbP
		1/4	*1/2*	*1/3*	*1/4*	*1/2*	*1/3*
QCIF 15Hz	***Foreman***	47.0	68.1	54.6	56.5	75.7	63.5
	Paris	49.5	69.5	57.4	59.1	77.0	65.6
	Tempete	44.9	65.3	53.8	58.2	75.5	64.6
QCIF 30Hz	***Foreman***	50.9	70.7	59.0	64.6	81.0	70.6
	Paris	52.6	71.7	61.1	66.0	82.0	71.4
	Container	65.6	78.1	73.0	79.9	87.2	85.2
CIF 30Hz	***Mobile***	46.6	66.1	55.9	62.2	78.2	68.5
	Paris	50.2	70.2	59.0	61.6	79.2	67.4
	Tempete	44.2	65.1	53.0	59.2	76.4	65.2
SD 25Hz	***Mobile***	46.3	65.8	54.2	69.2	81.7	73.3
	Parkrunner	47.0	68.1	57.0	64.0	79.7	70.0
Average Percent		49.5	68.9	58.0	63.6	79.4	69.5

Authorized licensed use limited to: Penn State University. Downloaded on April 01,2025 at 17:45:30 UTC from IEEE Xplore. Restrictions apply.